Spring 5-9-2015

Spatially Random Processes in One-Dimensional Maps: The Logistic Map and The Arnold Circle Map

An T. Le
University of Colorado Boulder, amy.le792@gmail.com

Spatially Random Processes in One-Dimensional Maps: The Logistic Map and The Arnold Circle Map

by

A. T. Le

B.S., University of Colorado, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Applied Mathematics

2015

This thesis entitled:
Spatially Random Processes in One-Dimensional Maps: The Logistic Map and The Arnold Circle Map
written by A. T. Le
has been approved for the Department of Applied Mathematics

_____

Dr. James Meiss

_____

Dr. Juan Restrepo

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Le, A. T. (M.S., Applied Mathematics)

Spatially Random Processes in One-Dimensional Maps: The Logistic Map and The Arnold Circle Map

Thesis directed by Dr. James Meiss

One way to model in-situ remediation of contaminated groundwater is to consider spatially random processes in nonlinear systems. Groundwater remediation often requires injecting an aquifer with treatment solution, where degradation reactions break down the toxins. As the treatment solution and contaminated water flow through the aquifer, their movement is limited by the types of sediment found in the aquifer, which act as spatial barriers to mixing. The onset of chaos in this system implies the two solutions are well mixed, and thus the contaminants are rendered inert. The spatially random processes explored in this thesis are meant to mimic the distribution of sediment in the aquifer. These processes were constructed using uniform random variables and normal random variables, and incorporate an exponentially decaying spatial correlation.

The three-dimensional model of the fluid flow in the aquifer has been simplified to an in-depth study of two one-dimensional maps: the logistic map and the Arnold circle map. Injection of the treatment solution in the aquifer may be thought of as the initial condition imposed on the map. Numerical simulations of the one-dimensional maps lay the groundwork for future studies of higher-dimensional systems.

Simulations indicate evidence of newly stabilized regions of the randomized logistic map, as well as a breakdown of symmetry and stable behavior in the Arnold circle map. The combination of bifurcation diagrams and Lyapunov exponents from the randomized logistic map lead us to hypothesize the spatially random process may stabilize the map in regions previously unstable. In the random circle map, analysis of the Arnold tongues, devil's staircases, and Lyapunov exponents suggest the random processes incur chaotic behavior in typically stable regions.

Dedication

To my mom, who has always told me, “Just do your best.”

Acknowledgements

For his extensive patience, support, and guidance, I am truly grateful to Jim Meiss. For being on my committee, I offer thanks to Jim Meiss, Juan Restrepo, and Will Kleiber. For his unwavering confidence in me and continued encouragement, I have sincere gratitude for Swamy Ananthanarayan.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Groundwater Contamination

Below ground, water found in the spaces between soil, sand and rock is called groundwater. It moves slowly through **aquifers** as a function of pressure (**hydraulic head**), **porosity**, aquifer thickness, and aquifer **transmissivity**. Aquifers are water-bearing layers of rock and sediments [15]. The pressure exerted on the groundwater is called the hydraulic head, and it is measured in units of height, such as feet or meters. It is the sum of the pressure head and the elevation head [15]. In other words, the level that water entering a confined well will stand at is determined by the hydraulic head. Porosity refers to the ratio of the empty spaces to the total volume of the sediment; fine-grained materials tend to have greater porosities than coarse-grained materials since they are better sorted [15]. Porosity is quantitatively tied to water flow through the aquifer by K, the hydraulic conductivity[1] (how easily liquids pass through the pores in the sediment). Materials with high hydraulic conductivity allow more water to move through the aquifer than materials with low hydraulic conductivity. Table 1.1 outlines typical porosity and hydraulic conductivity ranges for three kinds of materials: clay, sand, and gravel.

[1] Typically experimentally derived.

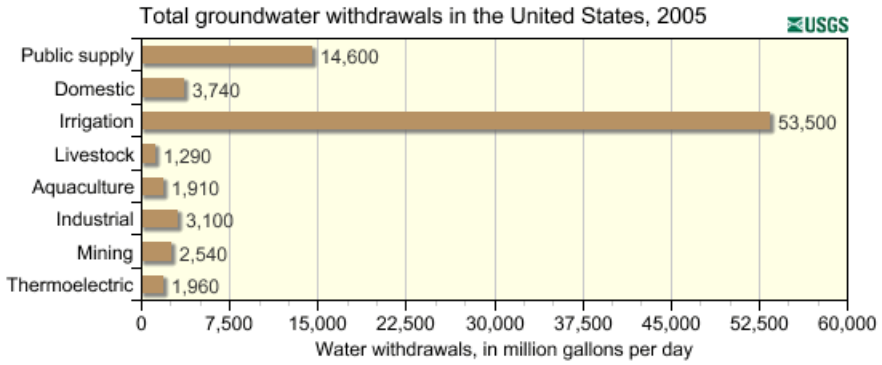
Table 1.1: Typical porosity and hydraulic conductivity ranges for clay, sand, and gravel [15] [7].

Grain Size	Material	Porosity	Hydraulic Conductivity K (m/s)
Fine	Clay	50%	$[5 \times 10^{-9}, 5 \times 10^{-6}]$
Medium	Sand	25%	$[10^{-8}, 10^{-6}]$
Coarse	Gravel	20%	$[5 \times 10^{-4}, 5 \times 10^{-2}]$

Transmissivity characterizes the volume of water flowing through a cross-section of the aquifer[2] . It is the product of hydraulic conductivity and the aquifer thickness. Hydraulic conductivity is a function of the hydraulic gradient, a dimensionless vector gradient between two or more hydraulic heads[3] . Essentially, a hydraulic gradient measures the rate of change of pressure per unit of distance at a given location in a specific direction.

Groundwater is vital to providing drinking water, irrigation, and makes up a large component in many industrial processes. Figure 1.1 demonstrates how groundwater is used in the United States. Most notably, about 51% of the total U.S. population and 99% of the rural population rely

Figure 1.1: The total groundwater withdrawals in the United States in 2005 is categorized in terms of use [19].



[2] The aquifer cross-section is 1 ft. by the aquifer thickness.

[3] The hydraulic gradient is equal to the difference between two hydraulic heads (ft) divided by the length between two piezometers, devices which measure the pressure of groundwater at a specific point

on groundwater as their drinking water source [5]. Unfortunately, by nature of being underground, groundwater is highly susceptible to contaminants, e.g. fertilizers, pesticides, road salt, gasoline, etc. Toxins that have leached into the water supply have deleterious effects on human health as well as serious environmental ramifications. In many cases, treatment requires injecting the aquifer with a treatment solution to break down the toxins. The treatment solution is only effective if it is thoroughly mixed with the contaminated water, so essentially this problem boils down to that of mixing, a favorite topic in nonlinear dynamics.

We explore the characteristics of spatially perturbed one-dimensional maps with the aim of understanding **in situ remediation** of contaminated groundwater. In situ remediation involves injecting a treatment solution in the groundwater to render the contaminants inert. Degradation reactions require the two solutions be in close proximity to each other; the solutions must mix. Engineered Injection and Extraction (EIE)[4] is an approach that uses sequential injection and extraction of water in wells surrounding the contaminated region to generate chaotic **advection**. A solute is advected through a system when it moves with the local fluid velocity [9]. Degradation reactions are more efficient when the solutions are thoroughly mixed together, so the onset of chaos in this system is a positive sign because it indicates the interface length between the two reactants is maximal, or at least large. The primary agent of mixing is the extent to which solutions move underground (transmissivity) [14].

Transmissivity, as a measure of how much water may flow horizontally, is the governing property of the degradation reactions because it limits the extent of chaotic advection. In turn, transmissivity depends on the type of sediment in the aquifer because the sediment porosity is directly proportional to hydraulic conductivity. The type of sediment found in the aquifer is highly variable, and different kinds of sediment hold and release water in various capacities. For instance, clay typically has very low hydraulic conductivity, but very high porosity (Table 1.1). This implies a section of the aquifer primarily made of clay can hold a large volume of liquid per volume of sediment, but liquid flows slowly through the region. In contrast, a section made of gravel would

[4] EIE sequences are typically heuristically developed to stretch and fold the fluid interface.

release water orders of magnitude faster. Without the aid of sophisticated equipment to determine the nature of the sediment underground, we might consider the sediment to be randomly distributed in space. Therefore, the spatial distribution of rocks and sediment plays an important role in the dynamics of the system.

Exploring the dynamics of a one-dimensional system lays the groundwork for understanding the dynamics of systems with higher dimensions. Two one-dimensional systems known to exhibit chaotic behavior are the logistic map and the circle map. Applying spatial perturbations to these maps and observing the subsequent dynamics may give an indication of what occurs in three-dimensional systems, like the groundwater model. A future work would generalize the results of this investigation to higher dimensions. In the one-dimensional maps, we aim to simulate transmissivity as a random function of space, and observations in the field suggest transmissivty follows a log-normal distribution [6]. In other words, the fluid flow in the aquifer is modeled as a one-dimensional map and the injection of the treatment solution in the aquifer is the initial condition imposed on the map.

1.2 Chaos

One-dimensional maps, such as the logistic map and the circle map, are analyzed in a variety of ways, e.g. fixed point iterations, cobweb diagrams, bifurcation diagrams, etc. This section offers some basic definitions and explanations of these concepts. One-dimensional maps are a subclass of dynamical systems in which time is discrete, rather than continuous. They take the form

$$x_{n+1} = f(x_n),$$

where $x_n \in \mathbb{R}$ or $x_n \in \mathbb{S}^1$, the circle. These maps demonstrate that even simple dynamical systems are capable of complex behavior, and as such, are also simple examples of chaos.

The sequence of iterates $x_0, x_1, ..., x_n$ in a map is called an **orbit**. As $n \to \infty$, orbits may converge to a fixed point, a set of periodic fixed points, or they may not converge at all.

A **fixed point** x^* of a function f is an element of the function's domain that is mapped to

itself by the function, i.e.

$$x^* = f(x^*).$$

In simple terms, a **periodic orbit** is an orbit that converges to a set of points after many iterates. Consider the repeating sequence $\{z_m\}$ in an arbitrary closed orbit $\{x_n\}$, where $m, n \in \mathbb{N}\,,\ \ m < n$. The sequence $\{z_m\}$ may contain many terms, but since $\{x_n\}$ is closed, $\{z_m\}$ eventually repeats. Thus, $z_{m+p} = z_m$ for some $p \in \mathbb{N},\ ,\ \ p \geq 1$. If this is true for any $m \in \mathbb{N}$, we call p the period of the sequence, and z_m the period p point. If this is true only for $m > m_0$, then the sequence $\{z_m\}$ is an eventually periodic orbit.

Fixed points and periodic orbits can be examples of attractors or repellors. An **attractor** has the following properties: it is an invariant set, it attracts an open set of initial conditions, and it is minimal [20]. Since the attractor is invariant, trajectories that begin in the attractor A cannot escape from A. By being minimal, we mean there does not exist a proper subset of A such that the previous two conditions are satisfied. An attractor is considered chaotic if it is sensitive to initial conditions. Otherwise, stable fixed points and periodic orbits are examples of attractors. Both types of attractors may have a **basin of attraction**, which is the set $S = \{x_0 : x(t) \to A, t \to \infty\}$, i.e. the largest set of initial conditions x_0 that are drawn to the attractor as time goes to infinity [20]. The opposite of an attractor is called repellor, a set that repels trajectories away from it.

Previously, we mentioned an attractor is considered chaotic if it exhibits sensitive dependence on initial conditions, but in fact, the definition of chaos is more complicated. Although no universal definition of chaos has been agreed upon, the following working definition is generally acceptable [20]:

Definition 1. *Chaos is aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions.*

(1) **Aperiodic long-term behavior** is another way of stating that there are trajectories within the map that do not limit to fixed points or, more generally, quasiperiodic orbits as $n \to \infty$.

(2) **Deterministic** is used to describe systems in which a given state has exactly one orbit, i.e. there is no random input or parameter. The observed behavior of the system arises from its nonlinearity, not from random noise.

(3) **Sensitive dependence on initial conditions** means that trajectories that start in almost the same place (ϵ apart) separate quickly. In most cases, this means the system has a positive Lyapunov exponent.

Lyapunov exponents are a way of quantifying the sensitive dependence on initial conditions for a specific orbit. If this limit exists, Lyapunov exponents are defined as

$$\lambda(x_0) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|. \tag{1.1}$$

For stable fixed points and stable periodic orbits, $\lambda(x_0) < 0$, and for chaotic attractors, $\lambda(x_0) > 0$. Equation 1.1 essentially states that if the magnitude of the derivative on the orbit is, on average, less than one (the logarithm becomes negative), then there is no chaos. However, if the expansion along the orbit is, on average, greater than one (the logarithm becomes positive), then it is a sign of chaos. $\lambda(x_0)$ is the same for all initial conditions x_0 in the same basin of attraction.

The existence and stability of fixed points in f are topics of great interest as they constitute a major part in understanding the dynamics of the map. If the domain is a complete metric space and f is a contraction mapping, the Contraction Mapping Theorem implies the map f has a unique fixed point [13]. We call a metric space X **complete** if every Cauchy sequence in X converges to an element of X. A sequence is Cauchy if the distance between any two elements of the sequence approaches zero as we consider more and more terms of the sequence. More formally, given a metric space X with metric ρ, a sequence $f_n \in X$ is **Cauchy** if $\forall \epsilon > 0\,,\ \exists N(\epsilon)$ such that whenever $m, n \geq N(\epsilon)\,,\ \rho(f_n, f_m) < \epsilon$ [13]. A metric measures the distance between two elements of the space f, g, and must satisfy the following three properties [13]:

(1) Positivity, $\rho(f, g) \geq 0$ and $\rho(f, g) = 0$ only when $f \equiv g$,

(2) Symmetry, $\rho(f, g) = \rho(g, f)$,

(3) Triangle inequality, $\rho(f,h) \leq \rho(f,g) + \rho(g,h)$.

A vector norm, such as the infinity norm, is an example of a metric.

Theorem 1. *The Contraction Mapping Theorem. Let $T : X \to X$ be a complete metric space X. If T is a contraction, i.e., if $\forall f, g \in X$, $\exists$ a constant $c < 1$ such that $\rho(T(f), T(g)) \leq c\rho(f,g)$, then T has a unique fixed point, $f^* = T(f^*) \in X$.*

The stability of the fixed point x^* is determined by perturbing the fixed point by η to see whether it is attracted to or repelled from x^*. A Taylor series expansion around the perturbation $x_{n+1} = x^* + \eta_{n+1}$ linearizes the map

$$\begin{aligned} x^* + \eta_{n+1} &= f(x^* + \eta_n) \\ &= f(x^*) + f'(x^*)\eta_n + \mathcal{O}(\eta_n^2), \\ \eta_{n+1} &= f'(x^*)\eta_n + \mathcal{O}(\eta_n^2). \end{aligned}$$

Neglecting the higher order terms in $\mathcal{O}(\eta_n^2)$ leaves $\eta_{n+1} = f'(x^*)\eta_n$. The multiplier is $\hat{\lambda} = f'(x^*)$. Solutions of the map are found by extending the recursion

$$\begin{aligned} \eta_1 &= \hat{\lambda}\eta_0 \\ \eta_2 &= \hat{\lambda}\eta_1 = \hat{\lambda}^2\eta_0 \\ &\ldots \\ \eta_n &= \hat{\lambda}^n\eta_0. \end{aligned}$$

If $|\hat{\lambda}| = |f'(x^*)| > 1$, $\lim_{n\to\infty} \hat{\lambda}^n = \infty$, and the fixed point x^* is **linearly unstable**. If $|\hat{\lambda}| = |f'(x^*)| < 1$, $\lim_{n\to\infty} \hat{\lambda}^n = 0$, and the fixed point x^* is **linearly stable**. If $|\hat{\lambda}| = |f'(x^*)| = 1$ the $\mathcal{O}(\eta_n^2)$ terms determine the local stability [20].

Likewise, a Taylor expansion about a periodic orbit demonstrates its stability criteria. Suppose we have a period 2 orbit. This implies $f(u) = v$ and $f(v) = u$. Both v, u are solutions of $f \circ f = f^2(x) = x$, so v, u are fixed points of $f^2(x)$. We compute the multiplier $\hat{\lambda}$,

$$\begin{aligned} \hat{\lambda} &= \frac{d}{dx} f(f(x)) \\ &= f'(f(x))f'(x). \end{aligned}$$

Evaluating the multiplier at $x = v$ yields

$$\hat{\lambda} = f'(u)f'(v).$$

By the symmetry of the final product, evaluating the multiplier at $x = u$ returns the same multiplier. Furthermore, we can generalize the stability analysis of a period p orbit. Each point of a periodic orbit $u_1, u_2, ...u_p$ is a fixed point of the map $f^p(x)$. Interestingly, any one of the points u_j may be a fixed point or a point on another periodic orbit of period less than p, e.g., a fixed point x^* of f is also the root of $f^p(x) - x$ for any p. We calculate the multiplier and evaluate along the points of the orbit, u_j, finding

$$\begin{aligned}\hat{\lambda} &= \frac{d}{dx}f^p(x) \\ &= \left(\frac{d}{dx}f^p(x)\right)\left(\frac{d}{dx}f^{p-1}(x)\right)...\left(\frac{d}{dx}f(x)\right) \\ &... \\ &= f'(u_p)f'(u_{p-1})...f'(u_2)f'(u_1).\end{aligned}$$

The magnitude of the multiplier can take on values

$$|\hat{\lambda}| = \prod_{j=1}^{p} |f'(u_j)| = \begin{cases} < 1 \\ = 1 \\ > 1 \end{cases}.$$

As in the fixed point scenario, $|\hat{\lambda}| < 1$ implies the orbit is stable, the $\mathcal{O}(\eta_n^2)$ terms of the Taylor expansion determine the local stability for $|\hat{\lambda}| = 1$, and $|\hat{\lambda}| > 1$ corresponds to an unstable orbit. However, the difference lies in the fact that the stability of the orbit depends on the derivative at all points on the orbit. In other words, if there is a point along the orbit where the slope is extremely large, then it is possible the orbit is unstable.

Linear stability is linked to nonlinear stability by the Hartman-Grobman Theorem [13]. The Hartman-Grobman theorem states the behavior of the linearized dynamical system near a fixed point is topologically equivalent to the behavior of the nonlinear system near the same point, as long as the multiplier $|\hat{\lambda}| \neq 1$. The term **topologically equivalent** means that there is a

homeomorphism, or a continuous and invertible map, between the linear and nonlinear system such that the directions of the trajectories are preserved [20]. Therefore, the results of linear stability analysis of fixed points and orbits translates to nonlinear stability.

Fixed points may be created, destroyed, or their stability may change. Such a qualitative change in dynamics is called a **bifurcation** [20]. These changes are strongly dependent on the parameters of the system. There are many types of bifurcations, such as [20]:

(1) saddle-node[5] : As a parameter increases or decreases, two fixed points approach one another, collide, and finally are destroyed. The reverse may also occur, where the system begins with no fixed points, then one is created, and finally it splits into two points.

(2) transcritical: No matter how a parameter varies, there is no creation nor destruction of fixed points. However, the stability of the fixed point may change between stable, semi-stable, and unstable. In all cases, it exchanges stability with another fixed point.

(3) pitchfork: Fixed points appear and disappear in symmetrical pairs as a parameter is varied. There are two types of pitchfork bifurcations:

   (a) supercritical pitchfork[6] : One stable fixed point splits into three points after the bifurcation: a pair of stable fixed points that flank one unstable fixed point, located where the original stable fixed point was.

   (b) subcritical pitchfork[7] : A pair of unstable fixed points exists only below the bifurcation. The pair also flanks one stable fixed point, which becomes unstable after the bifurcation.

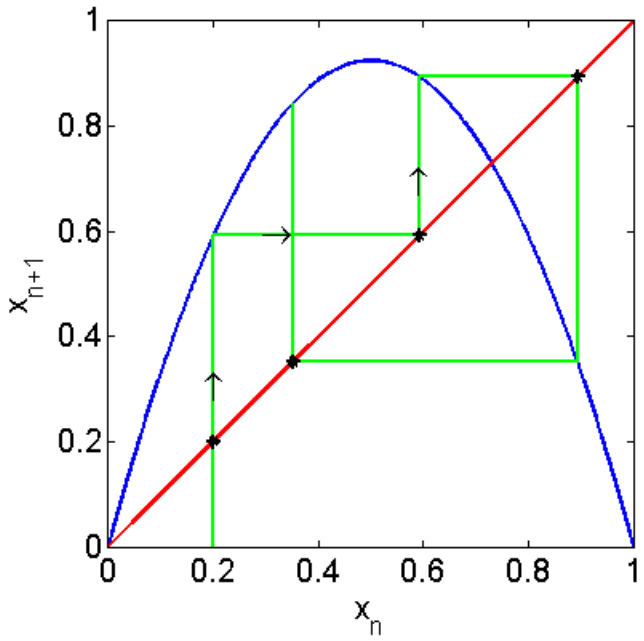
(4) period-doubling[8] : When a parameter increases past a certain point, a stable period $2p$ orbit appears near a period p orbit, and the period p orbit looses stability. One mechanism for the creation of chaos is if an infinite sequence of period-doubling bifurcations occurs.

[5] Also called a fold bifurcation, a turning-point bifurcation, or a blue-sky bifurcation.
[6] Sometimes referred to as a forward, soft or safe bifurcation.
[7] Can be called an inverted, backward, hard, or dangerous bifurcation.
[8] Also known as a flip bifurcation.

A bifurcation diagram is a visual demonstration of the change in dynamics of a system as its parameters are varied. Bifurcation diagrams are constructed by plotting the qualitative behavior of the orbits of a map as a function of its parameters, such as the locations of the stable and unstable orbits as a function of the parameters of the map.

A **cobweb diagram** is a graphical representation of the orbit iterations. The method of the cobweb diagram can be outlined in four steps, and is depicted in Figure 1.2.

Figure 1.2: A one-dimensional map (blue) and the line $x_{n+1} = x_n$ (red) with a few iterates of the cobweb diagram (green). Cobweb diagrams encapsulate the dynamics of the orbit at a glance.



(1) Begin at the Cartesian coordinate pair $(x_0, f(x_0))$, which may also be written as (x_0, x_1)

(2) Plot horizontally the distance from $(x_0, f(x_0)) = (x_0, x_1)$ to $(f(x_0), f(x_0)) = (x_1, x_1)$

(3) Plot vertically from $(f(x_0), f(x_0)) = (x_1, x_1)$ to $(f(x_0), f(f(x_0))) = (x_1, x_2)$

(4) Repeat steps 2-3 until convergence or divergence is determined (or perhaps the maximum number of iterations has been reached): $(x_n, x_{n+1}) \to (x_{n+1}, x_{n+1}) \to (x_{n+1}, x_{n+2})$

1.2.1 Logistic Map

The Logistic map is quadratic recursive equation that maps the domain $[0,1] \to [0,1]$. It is a commonly studied example in nonlinear dynamics and has applications in population modeling. Robert May popularized the logistic map in 1976 [12]. He demonstrated that even simple nonlinear maps could have complicated dynamics. There is one parameter in the expression, r, which can take any value in the range [0,4] so that [0,1] is an invariant set of the map. We define the map $f : [0,1] \to [0,1]$

$$x_{n+1} = f(x_n) = rx_n(1 - x_n). \tag{1.2}$$

For a fixed initial condition x_0 and r, the long term behavior of the map may be obtained by iteration, using (1.2). For example, Figure 1.3 demonstrates a stable period 4 orbit, whereas

Figure 1.3: Deterministic Logistic Map (blue) for $r = 3.2$. There is a stable period 4 orbit. The period is calculated by counting the number of crossings from the cobweb diagram (green) on the line $x_{n+1} = x_n$ (red). The transient iterations were removed to demonstrate the long-term behavior of the orbit.

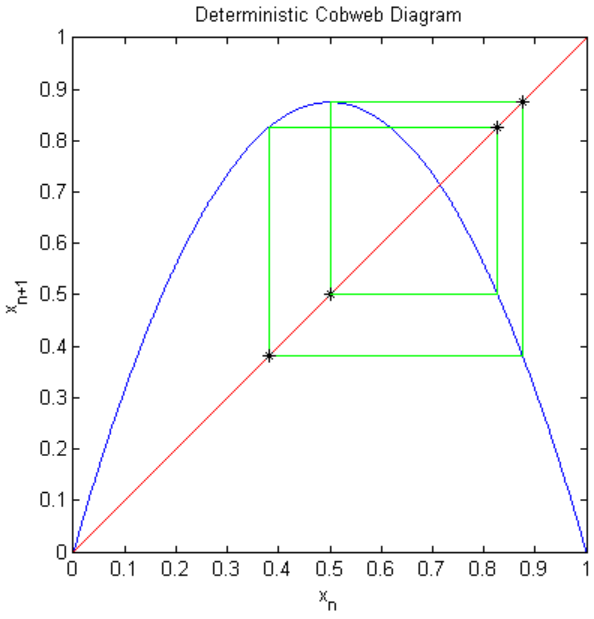


Figure 1.4 demonstrates a possibly chaotic orbit. The effects of the parameter r are tabulated in

Figure 1.4: Deterministic Logistic Map (blue) for $r = 3.8$. There appears to be no stable orbit. The cobweb diagram (green) behaves erratically, even after removing the transient behavior. For values of $r \in [3.5, 4]$, the system is chaotic.

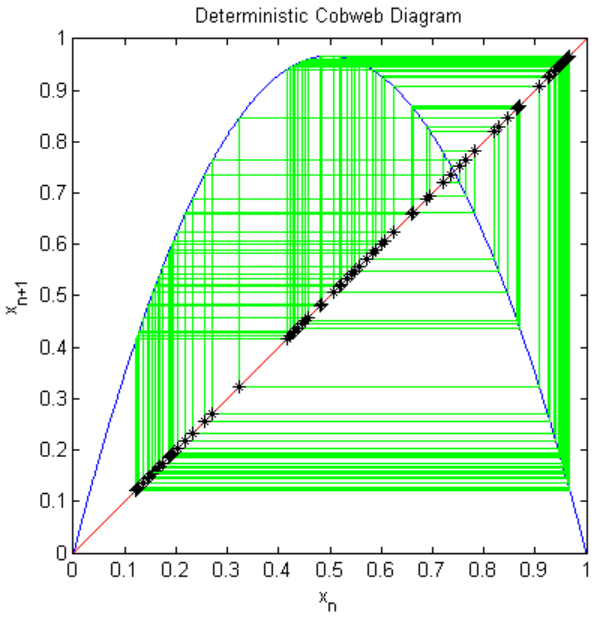


Table 1.2 and also graphically visualized in Figure 1.5, a bifurcation diagram. Most notably, there

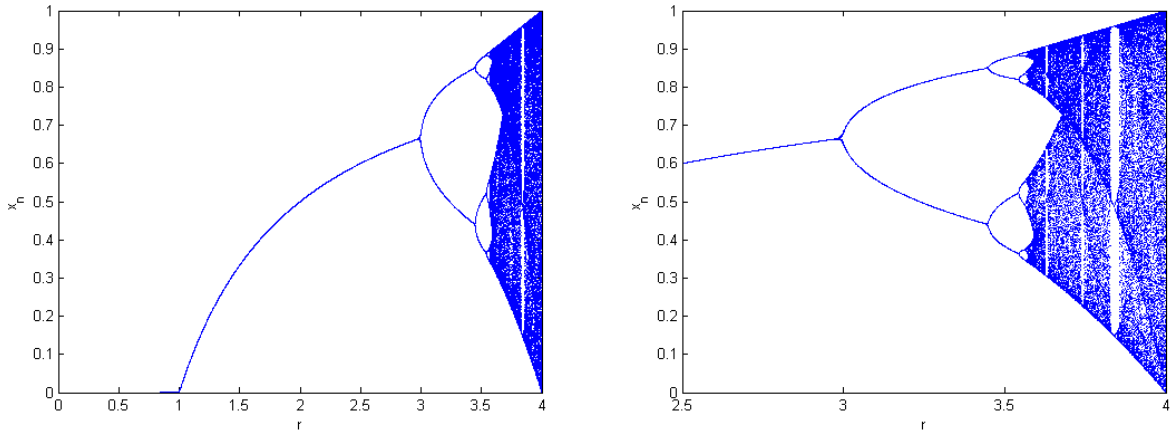
Table 1.2: As r is varied over [0,4], the logistic map undergoes notable changes in terms of stability [12].

$r \in [0, 1]$	convergence to the stable period 1 orbit, $x = 0$
$r \in [1, 2]$	convergence to the stable period 1 orbit, $x = \frac{r-1}{r}$
$r \in [2, 3]$	convergence to the stable period 1 orbit, $x = \frac{r-1}{r}$, but at a slower rate
$r \in [3, 3.44949]$	emergence of stable period 2 orbits
$r \in (3.44949, 3.54409)$	emergence of stable period 4 orbits
$r \in [3.54409, 3.56995)$	period doubling cascade
$r \approx 3.56995$	onset of chaos
$r \in (3.56995, 4]$	mostly chaotic behavior, but there are islands of stability (e.g. period 3 orbits for $r \approx 3.82843$)

is no chaos for values of $r < 3.5$, and for $r > 3.5$, the system often appears chaotic, although there are islands of stability interspersed throughout the bifurcation diagram. The Lyapunov exponent

λ of the logistic map is an indicator of chaos. Figure 1.6 demonstrates how the Lyapunov exponent of the logistic map changes sign as a function of r. λ approaches zero at the period-doubling bifurcations shown in Figure 1.5. The negative spikes in the plot around $r = 3.2$ and $r = 3.5$ correspond to the period 2 and period 4 orbits in the bifurcation diagram. In fact, $\lambda = -\infty$ when $f'(x_j) = 0$ for some x_j, and such orbits are called **superstable orbits**. Chaotic behavior appears where $\lambda > 0$, near $r = 3.57$, a point also known as the Feigenbaum period-doubling accumulation point. The islands of stability in the bifurcation diagram are visible where λ dips past zero, notably around $r = 3.83$.

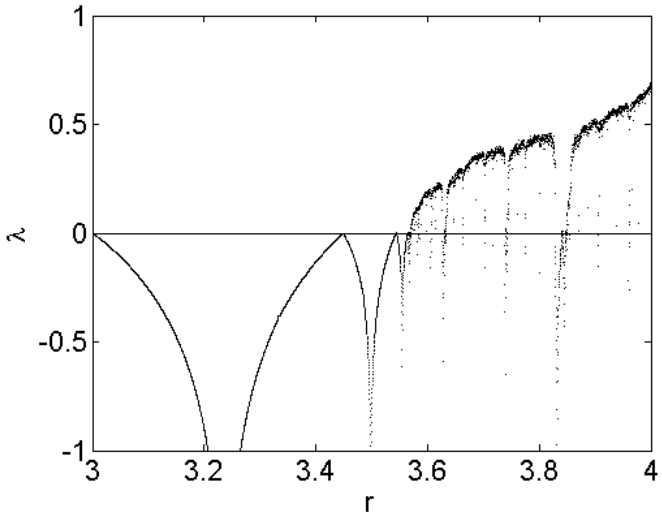
Figure 1.5: The behavior in Table 1.2 may be graphed in a bifurcation diagram, there the long term behavior of the map is a function of r. The left diagram ranges from $r \in [0, 4]$, and the right diagram ranges from $r \in [2.5, 4]$. Notice the islands of stability in the chaotic region of the map.



1.2.2 Circle Map

The circle map is another dynamical system whose deceivingly simple form gives way to complicated behavior. It was used to understand the dynamics of a kicked rotor by Vladimir Arnold, who is credited for discovery of the Arnold tongues, which are regions of stable periodic

Figure 1.6: The Lyapunov exponent for the deterministic logistic map, where $x_0 = 0.7$ for $r \in [3, 4]$. The positive values of λ for $r > 3.57$ indicate the system is chaotic.



motion in the bifurcation diagram of the map. Generally, a degree-one circle map takes the form

$$x_{n+1} = f(x) = x_n + \omega + g(x_n) \mod 1$$

$$g(x_n) = g(x_n + 1),$$

where the angle x has been normalized so that its range is $[0, 1)$ instead of $[0, 2\pi)$, ω is the natural frequency of the map, and $g(x)$ is the nonlinear component of the recursion [16]. The parameter ω represents the natural frequency of the rotor, and g models the driving force with period 1 in time. We define the circle S^1 as $\mathbb{R} \setminus \mathbb{Z}$. The **degree** of a map $f : S^1 \to S^1$ is the integer $\deg(f)$ given by

$$\deg(f) = F(1) - F(0),$$

where $F : \mathbb{R} \to \mathbb{R}$ is the lift of the map. F is called the **lift** of f if

$$w \circ F = f \circ w,$$

where $w : \mathbb{R} \to S^1$ is defined as

$$w(x) = e^{2\pi i x} = \cos(2\pi x) + i \sin(2\pi x).$$

w wraps $\mathbb{R}$ around the circle S^1 without critical points [4]. w is many-to-one, therefore it is not a topological conjugacy (homeomorphism) between F and f. When f is an orientation preserving diffeomorphism of the circle, the lift of f must be monotonic increasing, i.e. $F'(x) > 0$. A function is a **diffeomorphism** if it is a smooth, differentiable, invertible map between two or more manifolds. The term **topologically conjugate** means that there is a homeomorphism that transforms one map to another in such a way that the sense of time is preserved [20].

Definition 2. *Let $f : I \to J$. The function $f(x)$ is a homeomorphism if $f(x)$ is one-to-one, onto, and continuous, and $f^{-1}(x)$ is also continuous [4].*

Definition 3. *Let $f : I \to J$. The function $f(x)$ is a C^r-diffeomorphism if $f(x)$ is a C^r-homeomorphism such that $f^{-1}(x)$ is also C^r [4].*

Arnold introduced the two-parameter example

$$x_{n+1} = F(x_n) = x_n + \omega - \frac{k}{2\pi}\sin(2\pi x_n), \tag{1.3}$$

which we will explore as well. Equation (1.3) is the lift of the map f,

$$f(x_n) = x_n + \omega - \frac{k}{2\pi}\sin(2\pi x_n) \mod 1. \tag{1.4}$$

Taking the modulus of (1.3) maps the circle to itself. In both (1.3) and (1.4), there are two parameters: $k \in [0, \infty)$ is the coupling strength and $\omega \in [0, 1]$ is the natural frequency. A stable period 2 orbit is shown in Figure 1.7, and Figure 1.8 shows an orbit that appears to be quasiperiodic. When discussing f, we will deal exclusively with the lift, F.

The coupling strength k controls the amplitude of the oscillations in the circle map; no coupling is $k = 0$, and the coupling increases as $k \to \infty$. For $-1 \leq k < 1$, f is a diffeomorphism of S^1, and when $|k| = 1$, the map is a homeomorphism. f is monotone only if $|k| \leq 1$. For $k > 1$, the map is no longer one-to-one. In contrast, ω applies a positive vertical shift to the map, which causes the map to wrap around itself, due to the effects of the modulo operator.

A phenomenon of **mode-locking** may occur in the lift of the map, where after q iterations, the new angle differs from the initial value of x by exactly $p \in \mathbb{Z}$

$$x_{n+q} = x_n + p.$$

Figure 1.7: The cobweb diagram (green) for a deterministic circle map (blue) with $\omega = 0.5$ and $k = 1$. After 1000 iterations, we see a stable period 2 orbit. Transient iterations were removed.

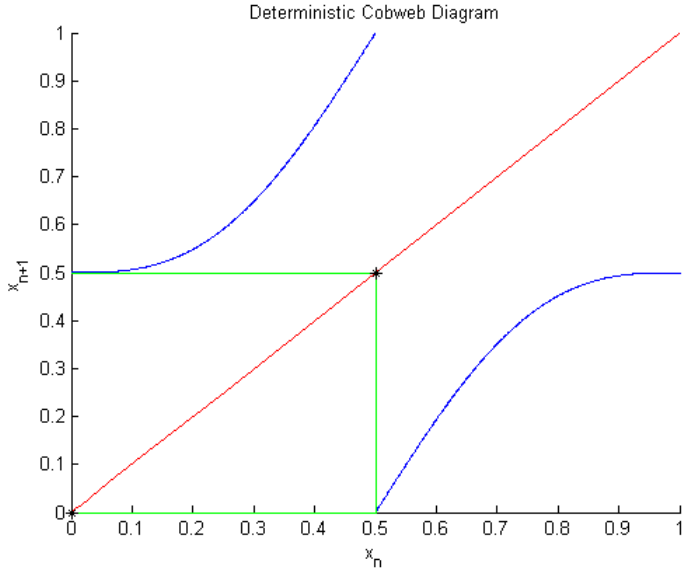
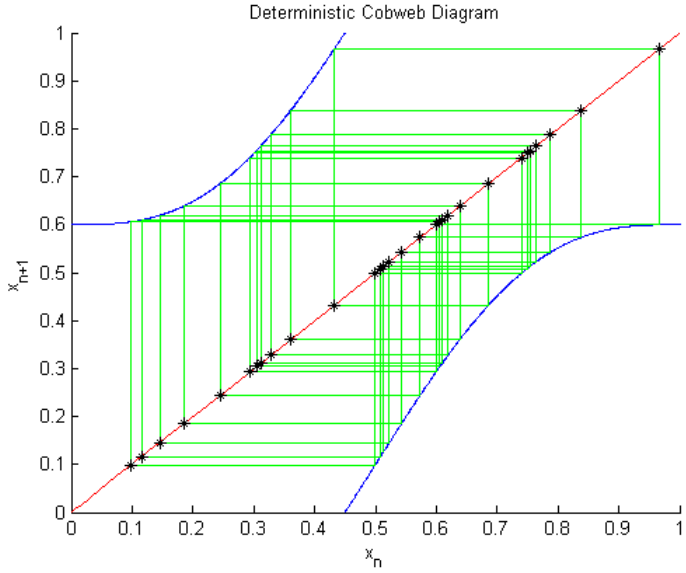


Figure 1.8: The cobweb diagram (green) for a deterministic circle map (blue) with $\omega = 0.6$ and $k = 1$. After 1000 iterations, we see a possibly quasiperiodic orbit. Transient iterations were removed.

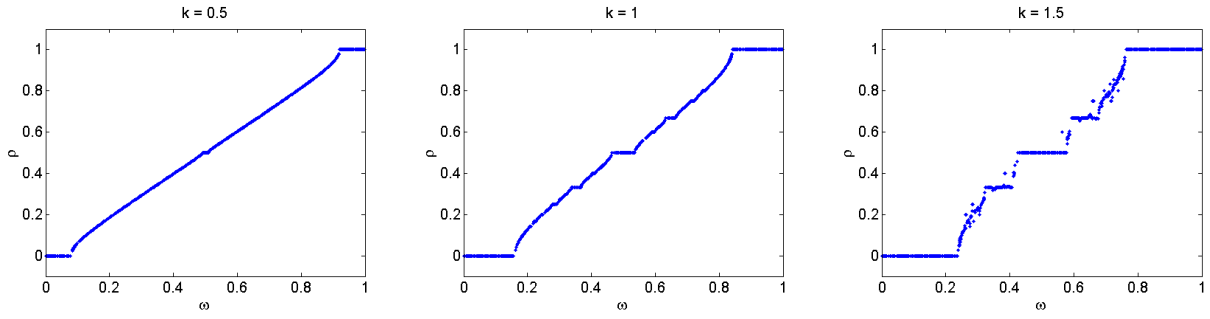


The mode-locked state of the lift F implies the orbit is periodic on $\mathbb{S}^1$, i.e. $x_{n+p} \mod 1 = x_n \mod 1$. The **rotation number** of the orbit is an important invariant of the circle map because it measures the average amount a point is rotated per iteration of the map. Essentially, it gives an idea of whether f has periodic orbits in $\mathbb{S}^1$.

Definition 4. *The rotation number of f, $\rho(f)$, is the fractional part of $\rho_0(F) = \lim_{n\to\infty} \frac{|F^n(x)|}{n}$ for any lift F of f. That is, $\rho(f)$ is the unique number in [0,1) such that $\rho_0(F) - \rho(f)$ is an integer [4].*

It is important that $\rho(f)$ depends only on f when f is a homeomorphism. But, if not, it can depend on x as well. Recall when $|k| > 1$, the map (1.3) is no longer one-to-one, so ρ may not exist [4]. When $\rho \in \mathbb{Q}$, the system is mode-locked. Since ρ is a rational number, it can be expressed as p/q, where $p, q \in \mathbb{Z}$. The order of the periodic orbit is q. If ρ does not exist (when f is not a homeomorphism), then the system may be in a chaotic state. When f is a homeomorphism and $\rho \in \mathbb{R} \setminus \mathbb{Q}$, the orbit is quasiperiodic. The changes in the rotation number ρ as ω is varied over [0,1] is graphically demonstrated as a devil's staircase in Figure 1.9. The devil's staircase is a monotone

Figure 1.9: The devil's staircase. For $|k| \leq 1$, f is a homeomorphism, so the staircase is monotone. The largest steps (where ρ appears constant) correspond with the simplest rationals. If $|k| > 1$, f is not a homeomorphism so ρ may not exist.



function that is constant at rational rotation numbers, and takes on every value in between the rational numbers [4]. Interestingly, the graph of $\rho(\omega)$ is a Cantor function. Cantor functions are everywhere continuous and monotone, but have zero derivative almost everywhere.

The theorem below establishes that the rotation number of the map $\rho(f)$ depends continu-

ously on f, and that if f is monotone, then all orbits have the same rotation number.

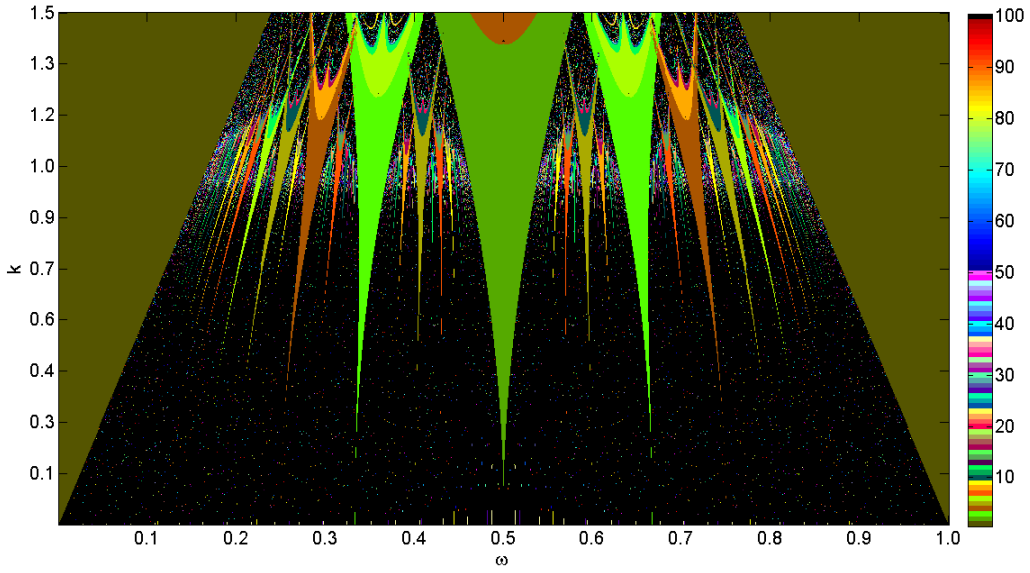
Theorem 2. *Let $f : S^1 \to S^1$ be an orientation-preserving diffeomorphism with lift F. Then*

$$\rho_0(F) = \lim_{n\to\infty} \frac{|F^n(x)|}{n}$$

exists and is independent of x and ρ is independent of the lift F. Hence the rotation number $\rho(f)$ is well-defined [4].

The width of each "step" in the devil's staircase corresponds to the width of the Arnold tongues (Figure 1.10) in the bifurcation diagram for k and ω in (1.3). Since there are two parameters in the circle map, its bifurcation diagram is two-dimensional. For any given (ω, k) pair, a pixel is plotted and colored according to the orbit period. The colored triangular regions in Figure 1.10 resemble tongues, and are aptly named as such. Recall the onset of chaos is marked by a positive

Figure 1.10: A color coded plot of the Arnold tongues in the circle map. This plot samples 1000 values of $\omega \in [0, 1]$ and $k \in [0, 1.5]$, and checks for periodicity up to period 100. The legend to the right demonstrates the period and corresponding color.



Lyapunov exponent. Figure 1.11 shows the Lyapunov exponent λ of the circle map as a function of ω for $k = 1.5$. Although small in magnitude, λ fluctuates between negative and positive values around

the Arnold tongues. The negative stretches in λ coincide with the stable behavior in Figure 1.10. Figure 1.12 demonstrates chaotic behavior occurs when $k > 1$ for a fixed $\omega = 0.7$.

Figure 1.11: The Lyapunov exponent for the deterministic circle map, where $k = 1.5, x_0 = 0.7$. The positive values of λ indicate the system may be chaotic. About 10,000 values of λ were computed over $\omega \in [0, 1]$.

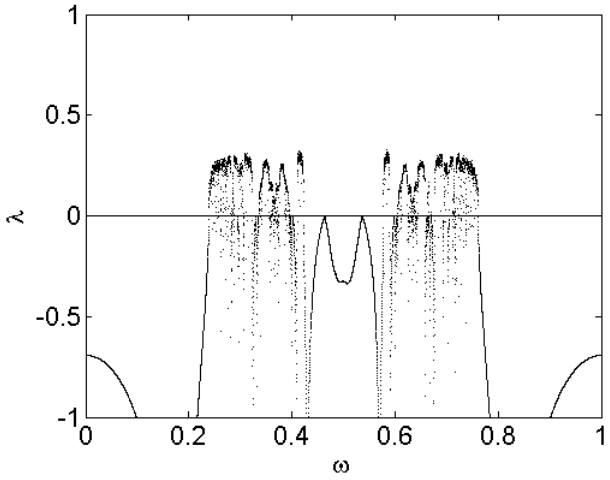
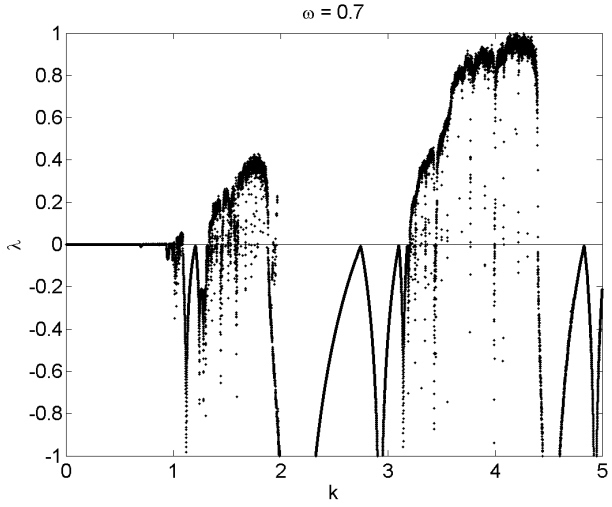


Figure 1.12: The Lyapunov exponent for the deterministic circle map, where $\omega = 0.7, x_0 = 0.7$. About 10,000 values of λ were computed over $k \in [0, 5]$.



Chapter 2

Spatially Random Maps

As studying the effects of random spatial perturbations on the dynamics of one-dimensional maps may yield implications for understanding the dynamics on higher dimensional maps, we present two case studies:

(1) the logistic map,

(2) the circle map.

A spatially random process is one that depends on space and is invariant in time; there is no time dependence in the process. A function that takes on a random value for any given location in space, but retains the same value if it were to return to this same location at a later time would be a spatially random process. The random spatial perturbations in both maps are intended to mimic the noise used in hydraulic modeling for transmissivity, which is assumed to be log-normal with an exponentially decaying spatial correlation. Observations in subsurface hydrology suggest flow parameters are log-normal [6]. The following sections offer background information on the nature of the spatially random processes and the characteristics of both maps with the spatial perturbations.

2.1 Spatially Random Processes

The noise in hydraulic modeling is often assumed to be log-normal with an exponentially decaying spatial correlation, so we let

$$\xi(x) = \ln(R(x)) \tag{2.1}$$

be a normal random variable that represents the noise. Then $R(x)$ is a log-normal random variable [6]. A normal distribution may be used for subsurface storage properties[1] , but in some cases, it is inappropriate because it would imply negative values have nonzero probability. Alternatively, a log-normal random variable may be used for physical parameters known to be nonnegative. Moreover, the log-normal distribution frequently appears when observing nonnegative flow parameters underground, like hydraulic conductivity and transmissivity [6]. An important property we would like to capture in this process is how the noise should vary based on position. In aquifers, porosity in a neighborhood of rock tends to be somewhat consistent (highly correlated), whereas porosity between two areas on opposite ends of the aquifer can be completely different. Thus, the noise in the random process should depend on the distance between two locations instead of the locations themselves, and we call this property homogeneity.

We begin to characterize the process by assuming the entire random process $\xi(x)$ is homogeneous, and defining the mean μ and variance σ^2 of $\xi(x)$ to be

$$\begin{aligned} \mu &= E[\xi(x)] = \ln(r) \\ \sigma^2 &= E[(\xi(x) - \mu)^2] = E[\xi(x)^2] - (\ln(r))^2. \end{aligned} \tag{2.2}$$

From probability, the definition of covariance is [18]

$$\begin{aligned} C(\xi(x), \xi(y)) &= E[(\xi(x) - E[\xi(x)])(\xi(y) - E[\xi(y)])] \\ &= E[\xi(x)\xi(y)] - E[\xi(x)]E[\xi(y)]. \end{aligned}$$

[1] Some examples of subsurface storage properties include porosity and moisture content.

Let us compute $C(\xi(y+x),\xi(y))$. Homogeneity implies $E[\xi(x)] = E[\xi(y)] = E[\xi(y+x)]$.

$$\begin{aligned} C(\xi(y+x),\xi(y)) &= E[(\xi(y+x) - E[\xi(y+x)])(\xi(y) - E[\xi(y)])] \\ &= E[(\xi(y+x) - \ln(r))(\xi(y) - \ln(r))]. \end{aligned} \tag{2.3}$$

As the entire process is homogeneous, the covariance is also homogeneous. Indeed, if the covariance function is defined as a function of x, y, but the variability is independent of location in space; rather, dependent on the separation vector between the locations, we call the covariance function homogeneous [6]. Homogeneity is consistent with the noise in nonnegative flow parameters, which is thought to rely on the distance between two points instead of their precise locations. C is also called a correlation function, which describes how variables at different positions in space are related. It physically represents the degree of correlation between two consecutive instances in the process. Correlations are typically strongest when iterates are nearby, and they decay as the distance between two iterates increases, although random processes could violate this assumption.

We construct a random function $\xi(x)$ on $[0,1]$ that satisfies (2.2) with an infinite Fourier series,

$$\xi(x) = \ln(r) + \sum_{n\in\mathbb{Z}} \hat{\xi}_n e^{2\pi i n x}. \tag{2.4}$$

Desirably, the magnitudes of the Fourier modes $\hat{\xi}_n$ diminish as $n \to \pm\infty$, which implies the random log-fluctuations are not identically distributed. If we modify the variance of the modes to decay exponentially, we achieve convergence in the sum. To ensure (2.4) stays in the real plane, we impose the condition

$$\hat{\xi}_n^* = \hat{\xi}_{-n}, \tag{2.5}$$

where $\hat{\xi}_n = a_n + ib_n \in \mathbb{C}$ and the $*$ operator denotes the complex conjugate. Equation (2.4) assumes $a_0 = 0$ because we wish to maintain $E[\xi(x)] = \ln(r)$ when $n = 0$ (the spatially independent term a_0 will be absorbed into $\ln(r)$). Since the expectation of a sum is equal to the sum of expectations,

we find [18]

$$\begin{aligned} E[\xi(x)] &= E\left[\ln(r) + \sum_{n\in\mathbb{Z}} \hat{\xi_n} e^{2\pi i n x}\right] \\ &= \ln(r) + E\left[\sum_{n\in\mathbb{Z}} \hat{\xi_n} e^{2\pi i n x}\right] \\ &= \ln(r) + \sum_{n\in\mathbb{Z}} e^{2\pi i n x} E[\hat{\xi_n}]. \end{aligned}$$

If we choose the mean of the Fourier modes $\hat{\xi}_n$ to be zero, then the mean of the Fourier Series is zero, which preserves the expected value of $\xi(x)$ in (2.2). Therefore, we let

$$\begin{aligned} \hat{\mu}_n &= E[\hat{\xi}_n] = 0 \\ \hat{\sigma}_n^2 &= E[\hat{\xi}_n^2] - (E[\hat{\xi}_n])^2 = E[\hat{\xi}_n^2]. \end{aligned} \tag{2.6}$$

If we assume the modes $\hat{\xi}_n$ are independent, then the product of two modes n, m, where $m \neq -n$, have expectation

$$\begin{aligned} E[\hat{\xi}_n \hat{\xi}_m] &= E[\hat{\xi}_n] E[\hat{\xi}_m] \\ &= 0. \end{aligned}$$

However, if $m = -n$,

$$\begin{aligned} E[\hat{\xi}_n \hat{\xi}_{-n}] &= E[\hat{\xi}_n \hat{\xi}_n^*] \\ &= E[|\hat{\xi}_n|^2] \end{aligned}$$

which follows from (2.5). In other words,

$$E[\hat{\xi}_n \hat{\xi}_m] = \begin{cases} E[|\hat{\xi}_n|^2] & : m = -n \\ 0 & : m \neq -n. \end{cases}$$

Notice we may choose a function of n for $E[|\hat{\xi}_n|^2]$. Remember that if we choose a function that causes $\lim_{n\to\infty} \hat{\xi}_n \to 0$ fast enough, then the sum in (2.4) will converge. For notation, let the function $S : \mathbb{Z} \to \mathbb{R}$ be

$$S(n) = E[|\hat{\xi}_n|^2] = \hat{\sigma}_n^2. \tag{2.7}$$

Equation (2.7) is also called the spectral density function. We introduce the notion of the spectral density function $S(n)$ as the Fourier transform of the correlation function (2.3) [6]. From (2.6), it is clear the correlation function is a function of x,

$$\begin{aligned} C(\xi(x+y),\xi(y)) &= E[(\xi(y+x)-\ln(r))(\xi(y)-\ln(r))] \\ \xi(y)-\ln(r) &= \sum_n \hat{\xi}_n e^{2\pi i n y} \\ \xi(y+x)-\ln(r) &= \sum_m \hat{\xi}_n e^{2\pi i m(x+y)} \\ C(\xi(x+y),\xi(y)) &= E\left[\sum_n \sum_m \hat{\xi}_n \hat{\xi}_m e^{2\pi i m x} e^{2\pi i y(n+m)}\right] \\ &= \sum_n \sum_m E\left[\hat{\xi}_n \hat{\xi}_m e^{2\pi i m x} e^{2\pi i y(n+m)}\right], \end{aligned}$$

since if $m \neq -n$, then $E[\hat{\xi}_n \hat{\xi}_m] = 0$ and if $m = -n$, then $e^{2\pi i y(n+m)} = 1$, then

$$C(x) = E[|\hat{\xi}_n|^2 e^{-2\pi i n x}].$$

The Fourier transform of $C(x)$ demonstrates

$$\begin{aligned} \hat{C}(k) &= \int_0^1 C(x) e^{-2\pi i k x} dx \\ &= \sum_{m,n} \int_0^1 E[\hat{\xi}_n \hat{\xi}_m] e^{2\pi i m(x+y)} e^{2\pi i n y} e^{-2\pi i k x} dx \\ &= S(k). \end{aligned}$$

By nature of being defined as the Fourier transform of the covariance function, the spectrum represents a distribution of variance over frequency. In fact, for $x = 0$, (2.3) reduces to the definition of variance of $\xi(x)$, so

$$\begin{aligned} C(x) &= \sum_{n\in\mathbb{Z}} S(n) e^{2\pi i n x} \\ C(0) = \sigma^2 &= \sum_{n\in\mathbb{Z}} S(n). \end{aligned}$$

A simple spectrum that falls off quickly is

$$S(n) = \alpha e^{-L|n|}, \tag{2.8}$$

where $L > 0$ is the correlation length (and is fixed for each simulation). Values of L close to zero will result in a small, negative exponent, which consequently increases the spectrum. On the other hand, larger values of L cause the spectrum to decay faster. This choice of $S(n)$ corresponds to a correlation function

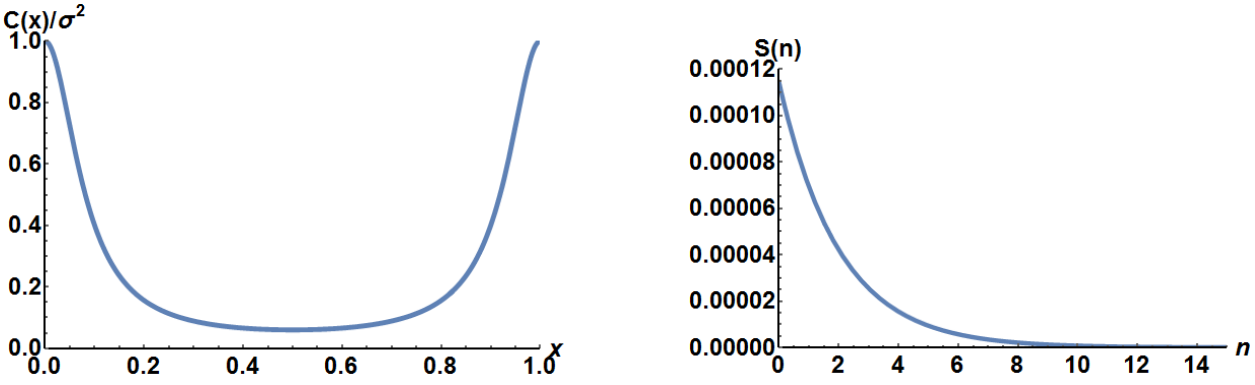
$$\begin{aligned}
C(x) &= \sum_{n\in\mathbb{Z}} S(n)e^{2\pi inx} = \sum_{n\in\mathbb{Z}} \alpha \frac{1}{e^{L|n|}} e^{2\pi inx} \\
&= \alpha + \sum_{n=1}^{\infty} \alpha e^{2\pi inx - L|n|} + \sum_{n=-1}^{-\infty} \alpha e^{2\pi inx - L|n|} \\
&= \alpha + \alpha \sum_{n=1}^{\infty} (e^{2\pi ix-L})^n + \alpha \sum_{m=1}^{\infty} (e^{-2\pi ix-L})^m \\
&= \alpha + \alpha \frac{e^{2\pi ix-L}}{1-e^{2\pi ix-L}} + \alpha \frac{e^{-2\pi ix-L}}{1-e^{-2\pi ix-L}} \\
&= \alpha \left(1 + \frac{-2e^{-2L} + 2\cos(2\pi x)e^{-L}}{e^{-2L} - 2\cos(2\pi x)e^{-L} + 1}\right).
\end{aligned}$$

Finally,

$$C(x) = \alpha \frac{e^{2L} - 1}{e^{2L} - 2\cos(2\pi x)e^{L} + 1}. \tag{2.9}$$

Figure 2.1 demonstrates an example of a covariance-spectrum pair from (2.9) and (2.8). In (2.8) and (2.9), the parameter α is a constant used to normalize $C(x)$. Recall from (2.3) when $x = 0$,

Figure 2.1: The correlation function $C(x)$ and the spectral density $S(n)$ for $L = 0.5, \sigma = 0.0216, \alpha = 0.000114$.

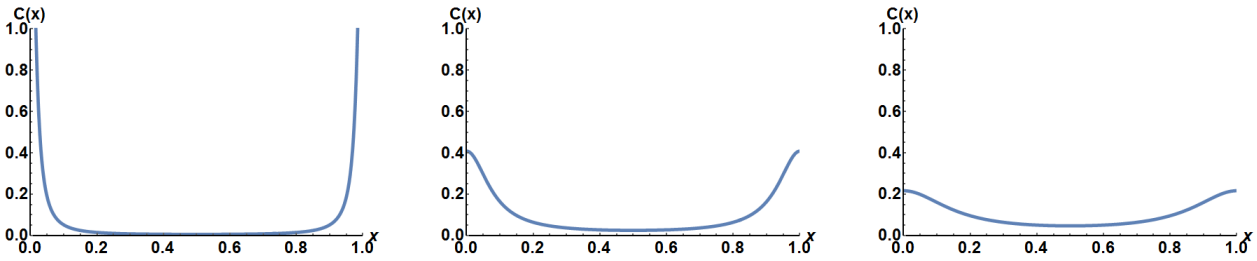


the covariance function reduces to the definition of variance, therefore

$$\begin{aligned}\sigma^2 &= \alpha\frac{e^{2L}-1}{e^{2L}-2\cos(0)e^L+1}\\ \alpha &= \sigma^2\frac{e^{2L}-2e^L+1}{e^{2L}-1}\\ \alpha &= \sigma^2\frac{(e^L-1)^2}{(e^L-1)(e^L+1)}\\ \alpha &= \sigma^2\tanh(L/2).\end{aligned} \tag{2.10}$$

The parameter α is now defined in terms of the variance σ^2 of $\xi(x)$. Examples of the function $C(x)$ are demonstrated in Figure 2.2 for various values of L.

Figure 2.2: The correlation function $C(x)$ for $L \in \{0.1, 0.5, 1\}$. For small L (leftmost graph), the correlation is stronger between iterates than for large L (rightmost graph).



We may wish to bound the distribution of the noise, especially if we apply the noise to the logistic map. Its parameter r must take on values in the interval $[0, 4]$, or else iterates of the map may leave $[0, 1]$. If we substitute $R(x)$ for the constant parameter r, then we require $R(x) \leq 4$. One way to prevent the possibility of exiting $[0, 1]$ is to bound the distribution of $\hat{\xi}_n$ from (2.4). Suppose the probability density function for $\hat{\xi}_n$ is nonzero only in the complex square centered at the origin with side length $2M_n$ (shown in Figure 2.3). Thus, $|a_n|, |b_n| \leq M_n$, and for a fixed r, we can bound $\ln(R(x))$ using (2.1) and (2.4). First, recall

$$\xi(x) = \ln(R(x)),$$

and because we want $R(x) \leq 4$, this implies

$$\xi(x) = \ln(R(x)) \leq \ln(4).$$

Since we also defined $\xi(x)$ as a series, we have

$$\xi(x) = \ln(R(x)) = \ln(r) + \sum_{n\in\mathbb{Z}} \hat{\xi_n} e^{2\pi i n x} \leq \ln(r) + \sum_{n\in\mathbb{Z}} |\hat{\xi_n}|,$$

where r is fixed for any given realization. Remember the sum in (2.4) was constructed to be real, so it is less than the sum over $|\hat{\xi}_n|$. If the sum over the magnitude of the Fourier modes is bounded by a constant, $\xi(x)$ will also be bounded by the same limit; therefore we choose

$$\begin{aligned} \ln(r) + \sum_{n\in\mathbb{Z}} |\hat{\xi_n}| &\leq \ln(4) \\ \sum_{n\in\mathbb{Z}} |\hat{\xi_n}| &\leq \ln(4/r). \end{aligned}$$

We have $|\hat{\xi_n}| = \sqrt{a_n^2 + b_n^2}$ and $|a_n|, |b_n| \leq M_n$, which implies

$$\begin{aligned} \sum_{n\in\mathbb{Z}} |\hat{\xi_n}| = \sum_{n\in\mathbb{Z}} \sqrt{a_n^2 + b_n^2} &\leq \sum_{n\in\mathbb{Z}} |a_n| + |b_n| \\ &\leq \sum_{n\in\mathbb{Z}} 2M_n. \end{aligned} \tag{2.11}$$

What remains is to choose M_n such that the sum in (2.11) is bounded by $\ln(4/r)$,

$$\sum_{n\in\mathbb{Z}} \hat{\xi_n} e^{2\pi i n x} \leq 2 \sum_{n\in\mathbb{Z}} M_n \leq \ln(4/r). \tag{2.12}$$

In order for this to occur, the sequence of side lengths M_n must be summable.

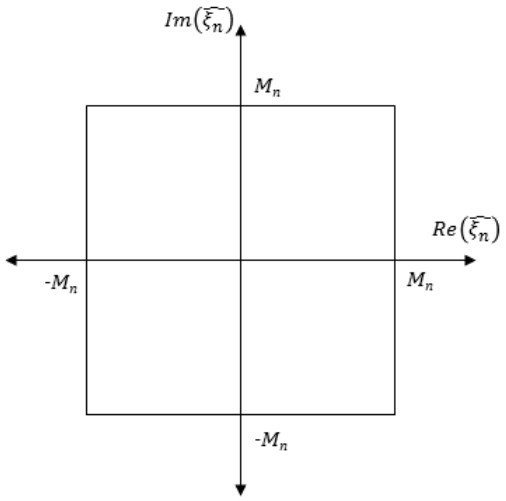
Using a bounded distribution to describe $\hat{\xi}_n$ will ensure $R(x) \leq 4$, unlike an unbounded distribution. Let $\hat{\xi}_n$ be uniformly distributed, so

$$h(\hat{\xi}_n) = h(a_n, b_n) = \begin{cases} \frac{1}{4M_n^2} & |a_n|, |b_n| \leq M_n \\ 0 & |a_n|, |b_n| > M_n \end{cases} \tag{2.13}$$

where $h : \mathbb{C} \to [0, 1] \in \mathbb{R}$ is the probability density function of $\hat{\xi}_n$. To find the relationship between $S(n)$ and M_n, recall that $S(n)$ is defined as the variance of the log-fluctuations (2.7). Then,

$$\begin{aligned} S(n) &= E[|\hat{\xi}_n|^2] = E[a_n^2 + b_n^2] \\ &= \frac{1}{4M_n^2} \int_{-M_n}^{M_n} \int_{-M_n}^{M_n} (a_n^2 + b_n^2)\, da_n\, db_n \\ &= \frac{2}{3} M_n^2 \\ M_n &= \sqrt{\frac{3}{2} S(n)}. \end{aligned} \tag{2.14}$$

Figure 2.3: The probability density function of $\hat{\xi_n}$ is uniformly distributed across this square region, centered at the origin. The square region has side length $2M_n$ and area $4M_n^2$.



Finally, using the expression for α (2.10), the expression for M_n (2.14), and the sum from (2.12), let

$$A = M_0 + 2\sum_{n=1}^{\infty} M_n \leq \ln(4/r).$$

Then, we find

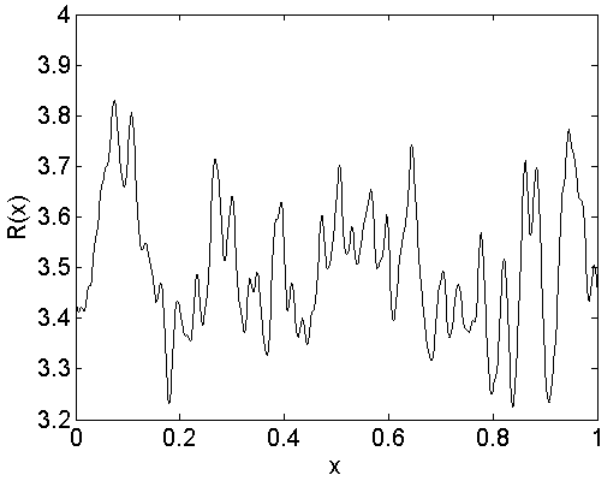
$$\begin{aligned} A &= \left(\sqrt{\frac{3}{2}\alpha} + 2\sqrt{\frac{3}{2}\alpha}\sum_{n=1}^{\infty} e^{-Ln/2}\right) \\ &= \sqrt{\frac{3}{2}\alpha}\left(1 + \frac{2e^{-L/2}}{1-e^{-L/2}}\right) \\ &= \sigma\sqrt{\frac{3}{2}}\sqrt{\tanh(L/2)}\left(\frac{e^{L/2}+1}{e^{L/2}-1}\right) \\ &= \sigma\sqrt{\frac{3}{2}}\frac{\sqrt{\tanh(L/2)}}{\tanh(L/4)} \leq \ln(4/r). \end{aligned}$$

Thus, the standard deviation σ of the variable $\xi(x)$ must be bounded from below by zero and from above by

$$0 < \sigma \leq \ln(4/r)\sqrt{\frac{2}{3}}\frac{\tanh(L/4)}{\sqrt{\tanh(L/2)}}. \tag{2.15}$$

Figure 2.4 is a sample realization of a function $R(x)$ that behaves according to (2.15).

Figure 2.4: The function $R : [0,1] \to [0,4]$ where $\hat{\xi}_n \sim Unif(-M_n, M_n), \sigma = 0.0386, L = 0.1, r = 3.5, N = 100$. N is the upper limit on the number of Fourier terms in the sum from (2.4).



The Lyapunov Central Limit Theorem demonstrates that the sum over independent, non-identical uniform random variables $\hat{\xi}_n$ tends to the normal distribution if certain conditions are upheld [3]. Unfortunately, we were unable to analytically demonstrate the limiting quantity in the theorem tends to zero. However, numerical simulations suggest the distribution of $\hat{\xi}_n$ is very similar to the normal distribution. To clarify, it remains to be determined whether random function $\xi(x)$ is normally distributed if $\hat{\xi}_n \sim Unif(-M_n, M_n)$. However, a histogram (Figure 2.5) of the quantity $\frac{T_m}{s_m}$ in (2.16) suggests there is some similarity to the standard normal distribution. Therefore, even though a spatially random process based on uniformly distributed Fourier mode amplitudes is not explicitly shown to be log-normal distributed, numerical simulations of a transformation of the process imply it mimics a log-normal distribution.

Theorem 3. *Let $\hat{\xi}_1, \hat{\xi}_2, ...\hat{\xi}_m$ be a sequence of independently distributed random variables, each*

having mean $E[\hat{\xi}_n] = \hat{\mu}_n$ *and variance* $\hat{\sigma}_n^2 < \infty$. *Define*

$$\begin{aligned} Y_n &= \hat{\xi}_n - \hat{\mu}_n \\ T_m &= \sum_{n=1}^{m} Y_n \\ s_m^2 &= \sum_{n=1}^{m} \hat{\sigma}_n^2. \end{aligned}$$

if $\exists\, \delta > 0$,

$$\lim_{m\to\infty} \frac{1}{s_m^{2+\delta}} \sum_{n=1}^{m} E[|Y_n|^{2+\delta}] \to 0,$$

then the distribution of

$$\frac{T_m}{s_m} \sim N(0,1).$$

That is, $\frac{T_m}{s_m}$ *tends to the standard normal.*

Based on the definitions in Theorem 3, we have for $\hat{\xi}_n \sim Unif(-M_n, M_n)$

$$\begin{aligned} Y_n &= \hat{\xi}_n - \hat{\mu}_n = \hat{\xi}_n \\ T_m &= \hat{\xi}_{-m} + \hat{\xi}_{-m+1} + ... + \hat{\xi}_0 + \hat{\xi}_1 + ... + \hat{\xi}_m \\ &= \hat{\xi}_1 + \hat{\xi}_1^* + ... + \hat{\xi}_m + \hat{\xi}_m^* \\ &= \sum_{n=1}^{m} 2a_n, \end{aligned}$$

since $\hat{\xi}_0 = 0$. A uniform random variable $X \sim Unif(c,d)$ has variance $Var(X) = \frac{1}{12}(d-c)^2$, so

$$\begin{aligned} Var(\hat{\xi}_n) = \hat{\sigma}_n^2 &= \frac{1}{12}(2M_n)^2 \\ &= \frac{1}{3}M_n^2 \\ &= \frac{1}{2}\alpha e^{-L|n|} \\ s_m^2 &= \frac{\alpha}{2}\left(e^{-L|-m|} + e^{-L|-m+1|} + ... + 1 + e^{-L} + ... + e^{-Lm}\right) \\ &= \frac{\alpha}{2} + \alpha\left(e^{-L} + ... + e^{-Lm}\right) \\ &= \frac{\alpha}{2} + \alpha\sum_{n=1}^{m} e^{-Ln}. \end{aligned}$$

We check whether the sum s_m^2 diverges as $m \to \infty$. After applying the Geometric series identity,

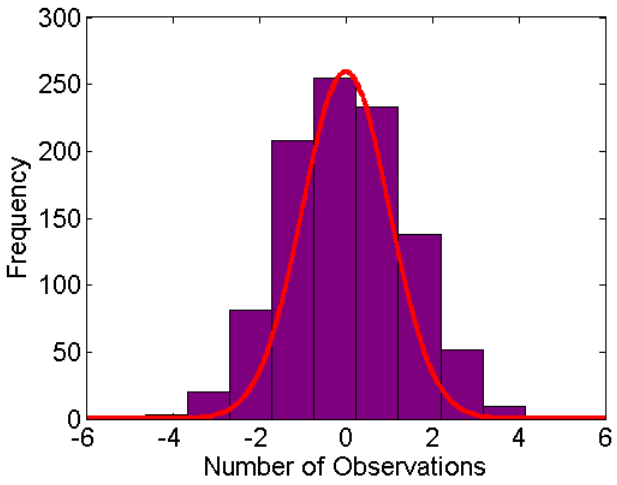
$$\begin{aligned}\lim_{m\to\infty} s_m^2 &= \frac{\alpha}{2} + \alpha \sum_{n=1}^{\infty} e^{-Ln} \\ &= \frac{\alpha}{2} + \left(\frac{\alpha}{1-e^{-L}} - \alpha \right) \\ &= \frac{\alpha(e^L+1)}{2(e^L-1)} \\ &= \frac{\alpha}{2\tanh(L/2)},\end{aligned}$$

we find the sum does indeed converge to a finite value. Using the expressions derived above, we find

$$\frac{T_m}{s_m} = \frac{\sum_{n=1}^{m} 2a_n}{\sqrt{\frac{\alpha}{2} + \alpha \sum_{n=1}^{m} e^{-Ln}}}. \tag{2.16}$$

A histogram in Figure 2.5 offers some insights on how the quantity in (2.16) is distributed.

Figure 2.5: A histogram of $\frac{T_m}{s_m}$ from (2.16), where $L = 0.1, N = 100, r = 3.3, \sigma = 0.0176$. The results of 1000 simulations are shown. The probability density function of the standard normal distribution is overlaid in red.



In summary, for each realization of the random map, the noise $\xi(x) = \ln(R(x))$ is a spatially random process whose mean value is $E[\xi(x)] = \ln(r)$. The Fourier modes of $\xi(x)$ are uniform random variables $a_n, b_n \sim Unif(-M_n, M_n)$, where $M_n = \sqrt{\frac{3}{2}S(n)}$. The function $S(n) = \alpha e^{-L|n|}$ represents

the spectral density, which was chosen to decay exponentially fast. The constant $\alpha = \sigma^2 \tanh(L/2)$ normalizes the corresponding correlation function $C(x)$. If we wish to restrict $R(x)$ to $[0, 4]$, then the standard deviation σ of the noise $\xi(x)$ must be bound according to (2.15). The log-normal property of the spatially random process may be lost if we choose $\hat{\xi}_n \sim Unif(-M_n, M_n)$.

2.2 Random Dynamics of the Logistic Map

The spatially random logistic map has not been as well explored in the literature as the temporally random case [1]. Athreya and Dai explore the concept of a time-varying logistic map, where the parameter r_i is a function of each time step, as opposed to position in space. At each time step, some $r_i \in \{r\}_0^\infty$ is applied to the map, where the sequence $\{r\}_0^\infty$ has independent and identically distributed random variables in [0,4]. In contrast, the spatially random logistic map replaces the parameter r from (1.2) with a random function of space, $R : [0, 1] \to [0, 4]$, and the random variables $\hat{\xi}_n$ are non-identical. In line with the deterministic map, $f : [0, 1] \to [0, 1]$, where

$$x_{n+1} = f(x_n) = R(x_n)x_n(1 - x_n). \tag{2.17}$$

Numerical simulations of (2.17) require a discrete limit on the sum in (2.4). We can combine the equations in (2.4) and (2.5) to arrive at the following form for $\xi(x)$,

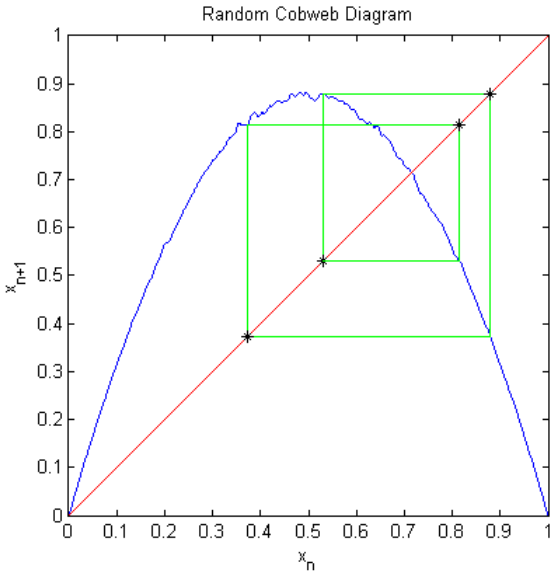
$$\xi(x) = \ln(r) + 2\sum_{n=1}^{N} a_n \cos(2\pi nx) - b_n \sin(2\pi nx), \tag{2.18}$$

where N represents the number of Fourier modes in the sum, and is an upper limit that we impose on the summation[2] . Figure 2.6 demonstrates one realization of a spatially random logistic map. Figure 2.7 depicts rough estimates[3] of upper and lower bounds for the random logistic map for various sets of parameters.

[2] In practice, $N \approx 10/L$, where L represents the correlation length. N should be chosen to be large enough such that the Fourier amplitudes may decay as more terms are included. This restriction will ensure the fluctuations are log-normal.

[3] The maxima and minima over 500 samples were recorded and plotted in these figures.

Figure 2.6: One instance of a random logistic map (blue), which was iterated for 1000 steps. The map dynamics have converged to a stable period 4 orbit (green). Notice the "wiggliness" in the parabola shape. $r = 3.5, L = 0.05, N = 200, \sigma = 0.0061$



2.3 Random Dynamics of the Circle Map

We will investigate a spatially random circle map of the form

$$x_{n+1} = F(x_n) = x_n + \Omega(x_n) - \frac{k}{2\pi}\sin(2\pi x_n), \tag{2.19}$$

where ω from (1.3) has been replaced with a function of space, $\Omega : S^1 \to S^1$. To distinguish between the logistic map and the circle map, we denote the function $R(x)$ in (2.1) as $\Omega(x)$; that is, $\Omega(x)$ is also meant to be a log-normal random variable. In Figure 2.8, a realization of the random circle map using the probability density function h from (2.13) is shown, where the orbit converges to a period 5 orbit. Figure 2.9, 2.10, and 2.11 depict rough estimates[4] of upper and lower bounds for the random circle map for various sets of parameters, where the random variables $a_n, b_n \sim Unif(-M_n, M_n)$.

Numerical simulations of (2.19) require a finite number of terms in the Fourier series of $\xi(x)$

[4] The maxima and minima over 500 samples were recorded and plotted in these figures.

Figure 2.7: A coarse demonstration (500 samples) of the upper and lower bounds of the random logistic map. Sample realizations are shown in red. From left to right: $\{\sigma = 0.1093, r = 2.7, L = 0.9, N = 112\}, \{\sigma = 0.0086, r = 3.5, L = 0.05, N = 200\}, \{\sigma = 0.0071, r = 3.7, L = 0.1, N = 100\}$

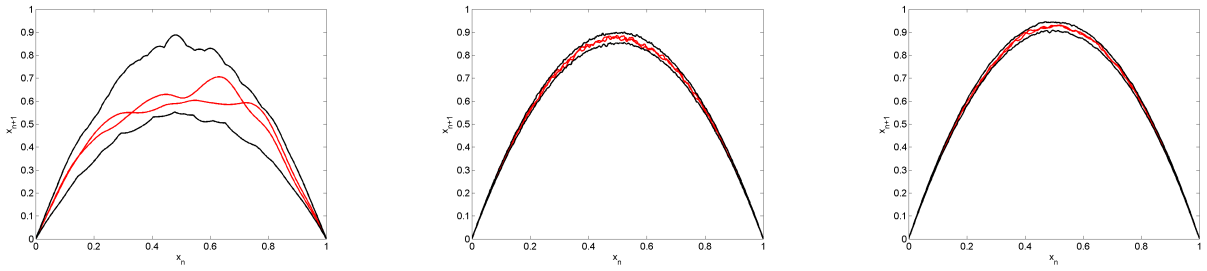
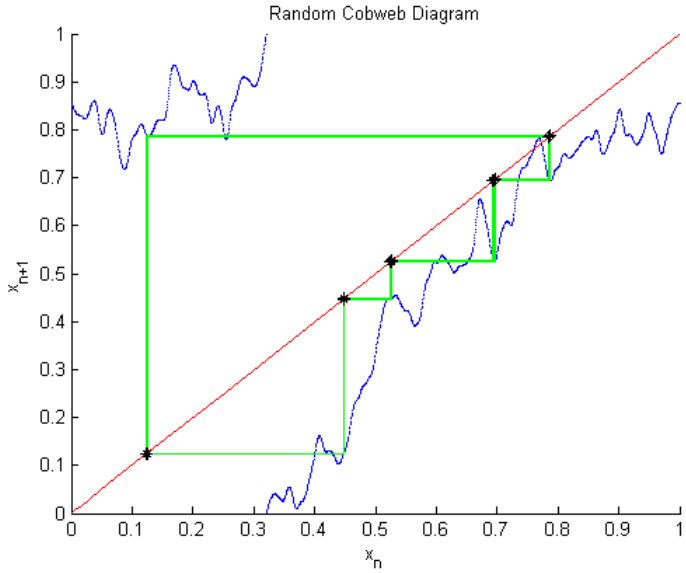


Figure 2.8: The cobweb diagram (green) for a random realization of the circle map (blue) with $\omega = 0.3, k = 1, \hat{\xi}_n \sim Unif(-M_n, M_n), \alpha = 10^{-5}, L = 0.1, N = 100$. The orbit has converged to a stable period 5 orbit. Its basin of attraction is S^1. Transient iterations were removed.



in (2.4). Therefore, we find

$$\xi(x) = \ln(\omega) + 2\sum_{n=1}^{N} a_n \cos(2\pi n x) - b_n \sin(2\pi n x). \tag{2.20}$$

However, since we do not require $\Omega(x)$ of the circle map (2.19) to be bounded as strictly as $R(x)$ of the logistic map (2.17), α may be chosen freely, and $h(\hat{\xi}_n)$ is not limited to the bounded uniform

Figure 2.9: A coarse demonstration (500 samples) of the upper (blue) and lower (black) bounds of the random circle map, where $\hat{\xi}_n \sim Unif(-M_n, M_n), k = 1, \omega = 0.3, \alpha = 10^{-5}, N = 100$. Sample realizations are shown in red. From left to right: $L = 0.9, L = 0.5, L = 0.1$.

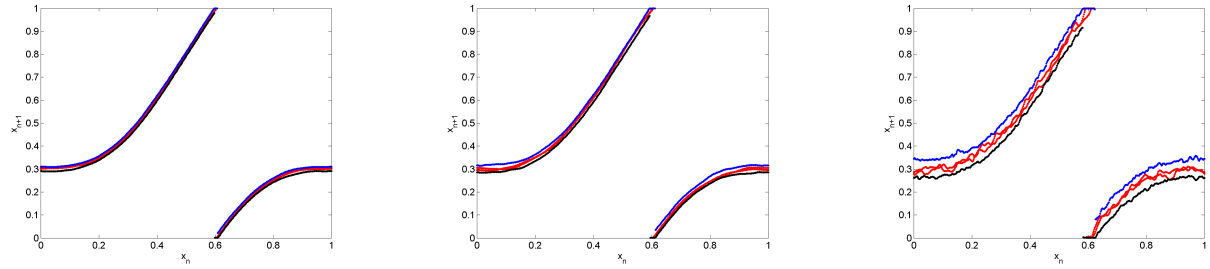


Figure 2.10: A coarse demonstration (500 samples) of the upper (blue) and lower (black) bounds of the random circle map, where $\hat{\xi}_n \sim Unif(-M_n, M_n)$ and $k = 1.5, \omega = 0.3, \alpha = 10^{-5}, N = 100$. From left to right: $L = 0.9, L = 0.5, L = 0.1$.

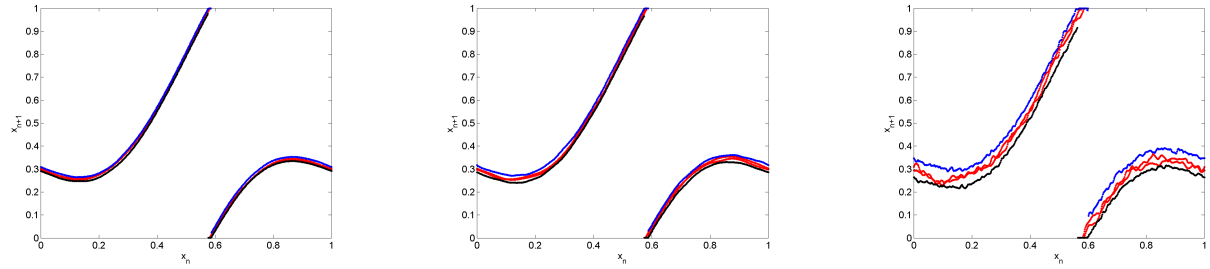
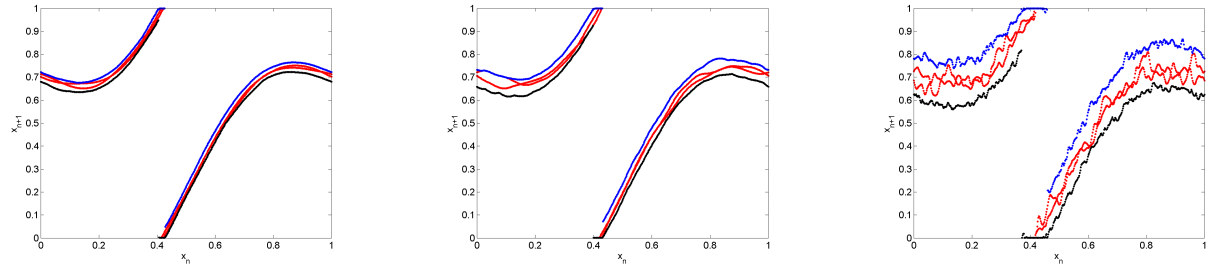


Figure 2.11: A coarse demonstration (500 samples) of the upper (blue) and lower (black) bounds of the random circle map, where $\hat{\xi}_n \sim Unif(-M_n, M_n), k = 1.5, \omega = 0.7, \alpha = 10^{-5}, N = 100$. From left to right: $L = 0.9, L = 0.5, L = 0.1$.



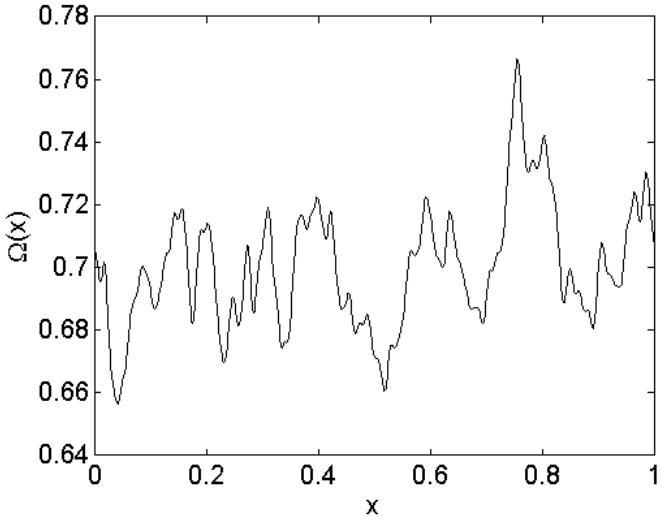
distribution. The choice of $h(\hat{\xi}_n)$ should be chosen such that the sum over the random variables is normal, as we aim to mimic log-normal noise.

For example, consider choosing $\hat{\xi}_n$ according to a normal distribution, with mean $\hat{\mu}_n = 0$ and variance $\hat{\sigma}_n^2 = S(n) = \alpha e^{-L|n|}$ for some $\alpha > 0$. In other words,

$$\hat{\xi}_n \sim N(0, \alpha e^{-L|n|}). \tag{2.21}$$

Figure 2.12 is a plot of an example of the spatially random process, $\Omega(x)$. Figure 2.13 demonstrates

Figure 2.12: The function $\Omega : [0,1] \to [0,4]$ where $\alpha = 10^{-5}, L = 0.1, \omega = 0.7, N = 100$, and $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$.



a realization of a random circle map, where the noise is drawn from the normal distribution. Figure 2.14, 2.15, and 2.16 depict rough estimates[5] of upper and lower bounds for the random circle map for various sets of parameters, where the random variables $a_n, b_n \sim N(0, \alpha e^{-L|n|})$.

The sum of m independent normal random variables $X_1, ..., X_m$ with means $\mu_1, ..., \mu_m$ and variances $\sigma_1^2, ..., \sigma_m^2$ has the convenient property

$$\sum_{n=1}^{m} X_n \sim N\left(\sum_{n=1}^{m} \mu_n, \sum_{n=1}^{m} \sigma_n^2\right).$$

[5] The maxima and minima over 500 samples were recorded and plotted in these figures.

Figure 2.13: The cobweb diagram (green) for a random realization of the circle map (blue) with $\omega = 0.3, k = 1, \alpha = 10^{-5}, L = 0.1, N = 100, \hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$. The orbit has converged to a stable period 4 orbit.

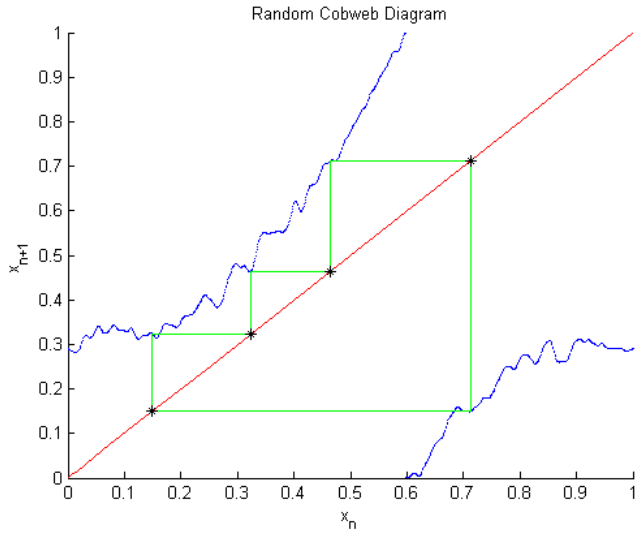


Figure 2.14: A coarse demonstration (500 samples) of the upper (blue) and lower (black) bounds of the random circle map, where $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|}), k = 1, \omega = 0.3, \alpha = 10^{-5}, N = 100$. Sample realizations are shown in red. From left to right: $L = 0.9, L = 0.5, L = 0.1$.

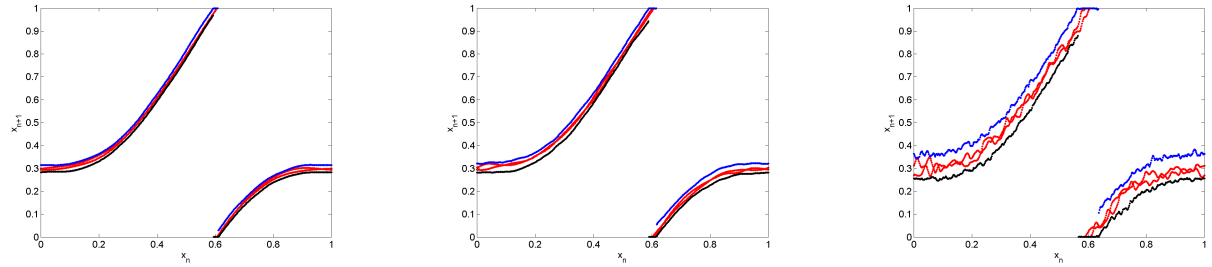


Figure 2.15: A coarse demonstration (500 samples) of the upper (blue) and lower (black) bounds of the random circle map, where $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|}), k = 1.5, \omega = 0.3, \alpha = 10^{-5}, N = 100$. From left to right: $L = 0.9, L = 0.5, L = 0.1$.

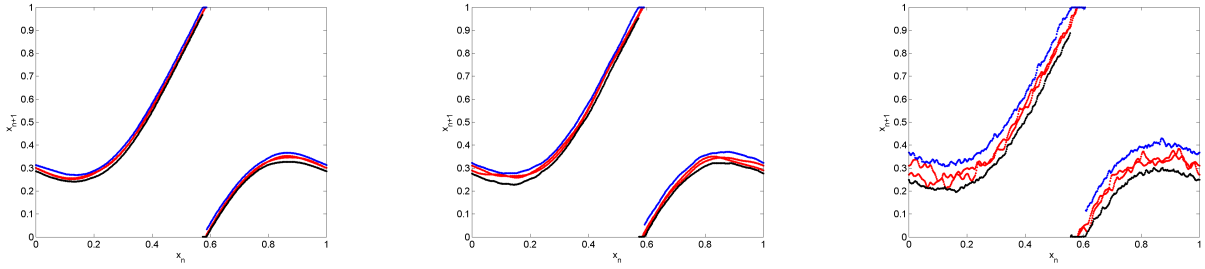
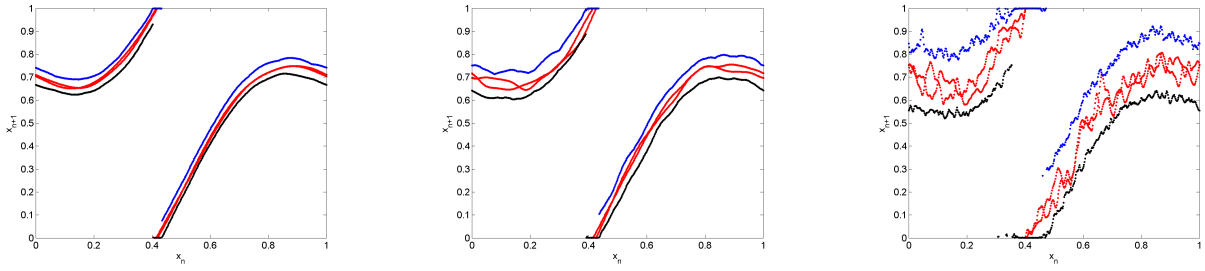


Figure 2.16: A coarse demonstration (500 samples) of the upper (blue) and lower (black) bounds of the random circle map, where $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|}), k = 1.5, \omega = 0.7, \alpha = 10^{-5}, N = 100$. From left to right: $L = 0.9, L = 0.5, L = 0.1$.



Thus, for $\hat{\xi}_{-N}, ..., \hat{\xi}_N$, we have

$$\sum_{n=-N}^{N} \hat{\xi}_n \sim N\left(0, \alpha \sum_{n=-N}^{N} e^{-L|n|}\right). \tag{2.22}$$

This implies $\xi(x)$ in (2.20) is also normal because it is a transformation of the sum in (2.22). Consequently, the noise in the circle map $\Omega(x)$ is log-normal, as desired.

Overall, we have a much simpler set of equations for the circle map than for the logistic map since $\Omega(x)$ is unrestricted. For each realization of the map, we choose a set of random variables a_n, b_n that correspond to the Fourier modes of $\xi(x)$, where $\hat{\xi}_n = a_n + ib_n$. The variables a_n, b_n may be drawn from an unbounded distribution, as long as the variance of Fourier modes $\hat{\sigma}_n^2$ decays according to the spectral density $S(n)$ in (2.7).

Chapter 3

Results

3.1 Implementation of Randomness in the Logistic Map

We explored how the random logistic map changes in terms of its bifurcation diagram and Lyapunov exponents, and also the distribution of period p orbits in terms of a set of histograms. The standard deviation of $\xi(x)$ was simulated for two values: the maximum value and half of the maximum value from (2.15). Scaling σ affects the variance $\hat{\sigma}_n^2$ of the Fourier modes of $\xi(x)$ because there is σ dependence in the expression for $\hat{\sigma}_n^2$ when $\hat{\xi}_n \sim Unif(-M_n, M_n)$,

$$\begin{aligned}\hat{\sigma}_n^2 &= S(n) = \alpha e^{-L|n|} \\ &= \sigma^2 \tanh(L/2) e^{-L|n|}.\end{aligned}$$

An unexpected result of the simulations is the presence of the stable periodic orbits in the right side of the bifurcation diagram (Figure 3.1, 3.2, and 3.3). These stable orbits appear for both the case where the standard deviation σ of $\xi(x)$ from (2.15) is chosen to be the maximal value and the case where we choose half of the maximal value. Bifurcation diagrams are shown in Figure 3.1 and Figure 3.2, which correspond to choosing σ to be the maximum, and in Figure 3.3, where σ is half of the maximum. Specifically, this stable region occurs between $r \in [3, 4]$, and is most dominantly featured for the bifurcation diagrams with $L = 0.1$. For the case where $\sigma = \sigma_{max}$, Figure 3.2 zooms in on the range $r \in [3, 4]$ to show the newly stabilized region more clearly. In the deterministic case (Figure 1.5), this region was previously unstable, except for windows of stable orbits.

Surprisingly, the stable low period orbits for $r \in [3, 4]$ endure even when the noise is restricted by reducing σ. Further, it appears the density of stable orbits for $r < 3$ diminishes as L is increased for both values of σ. One difference is that there are fewer stable high period orbits when σ is small, which makes sense if you reduce the amount of noise in the spatially random process.

In general, increasing L in the bifurcation diagram for the random logistic map (Figure 3.1) increases the spread of orbit locations for small r, and it may also result in fewer stable orbits for small r, due to the low density of points in the diagram. Increasing L also obscures the previously mentioned newly stable region with high-period orbits.

In the bifurcation diagrams (Figure 3.1, 3.3), each value of $r \in [0, 4]$ was tested for N_{x_0} initial conditions for $x_0 \in [0, 1]$. The discretization of r for the simulation is represented by the value N_r, the number of subintervals. If iterating the map for any given set of parameters resulted in finding no periodic orbit of period p_{max} or less, then no orbit is recorded in the bifurcation diagram. Otherwise, the orbit locations are plotted along the y-axis in the figures and color coded according to period. An orbit was denoted period p if, after iterating the map 1,000 times, $x_p = x_{n+p}$ within a tolerance of $\epsilon = 10^{-6}$.

Figure 3.1: Bifurcation diagrams of the random logistic map, where $r \in [0, 4]$, $\Delta r = 0.002$, $N = 100$, the number of initial conditions tested is N_{x_0}, $L \in \{0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$, and $\sigma = \sigma_{max}$. Plots are read left to right, and top to bottom. Number of simulations is 1.8 million. The legend shows the color for each period. Orbits up to period 256 were checked.

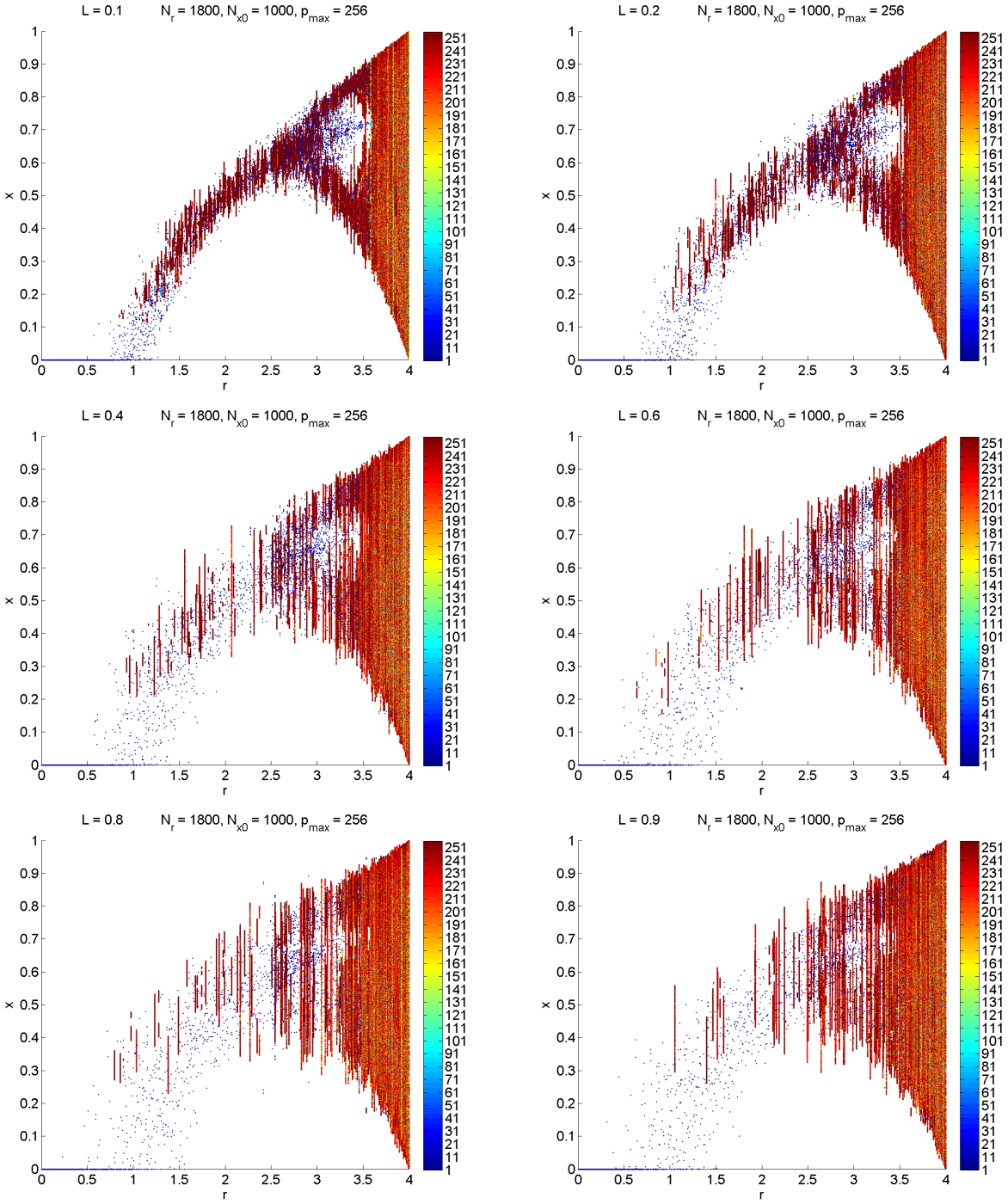


Figure 3.2: A zoomed in view of Figure 3.1, where $r \in [3, 4]$, $\Delta r = 0.001$, $N = 100$, the number of initial conditions tested is N_{x_0}, $L \in \{0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$, and $\sigma = \sigma_{max}$. Plots are read left to right, and top to bottom. Number of simulations is 1.8 million. The legend shows the color for each period. Orbits up to period 256 were checked.

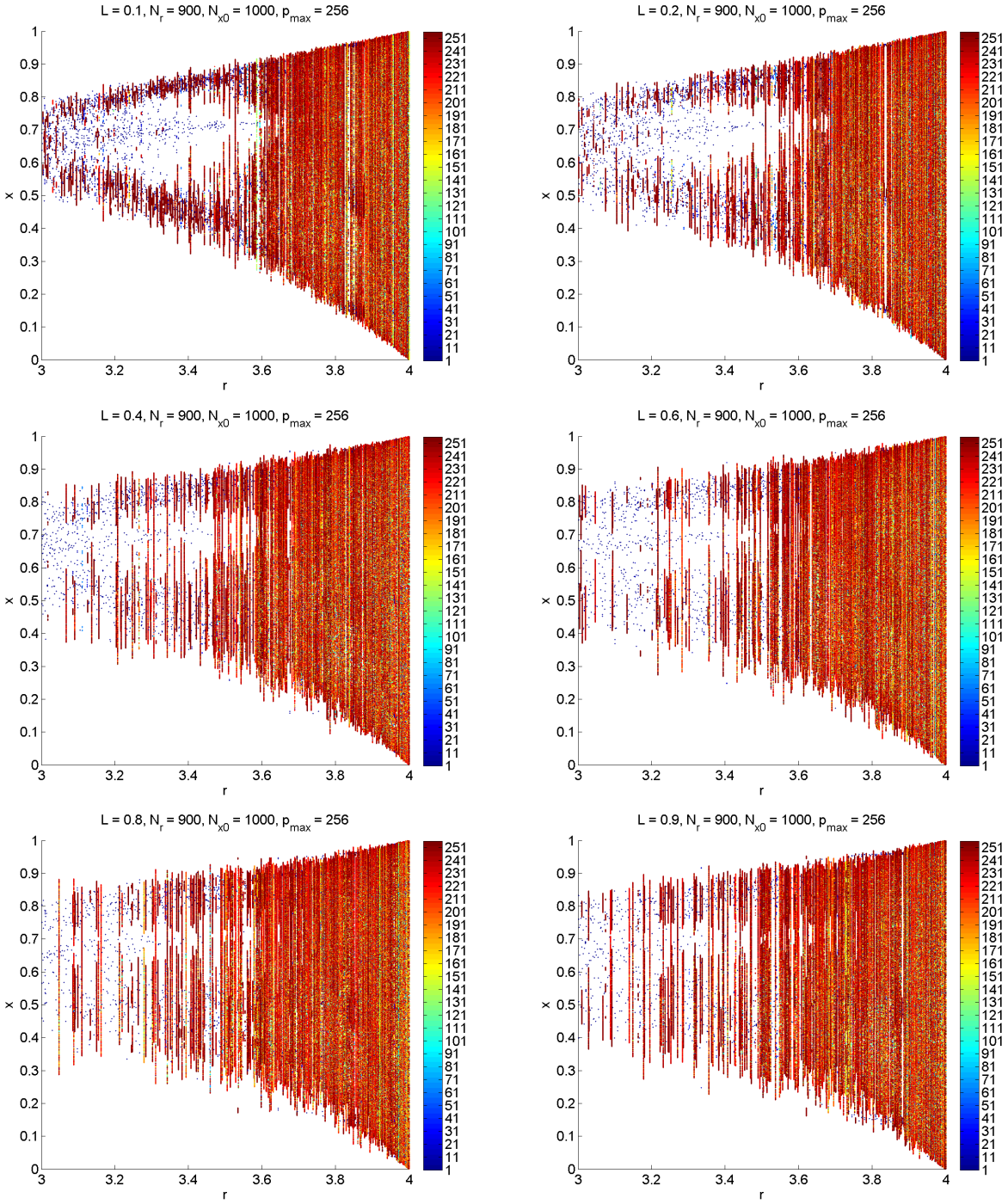
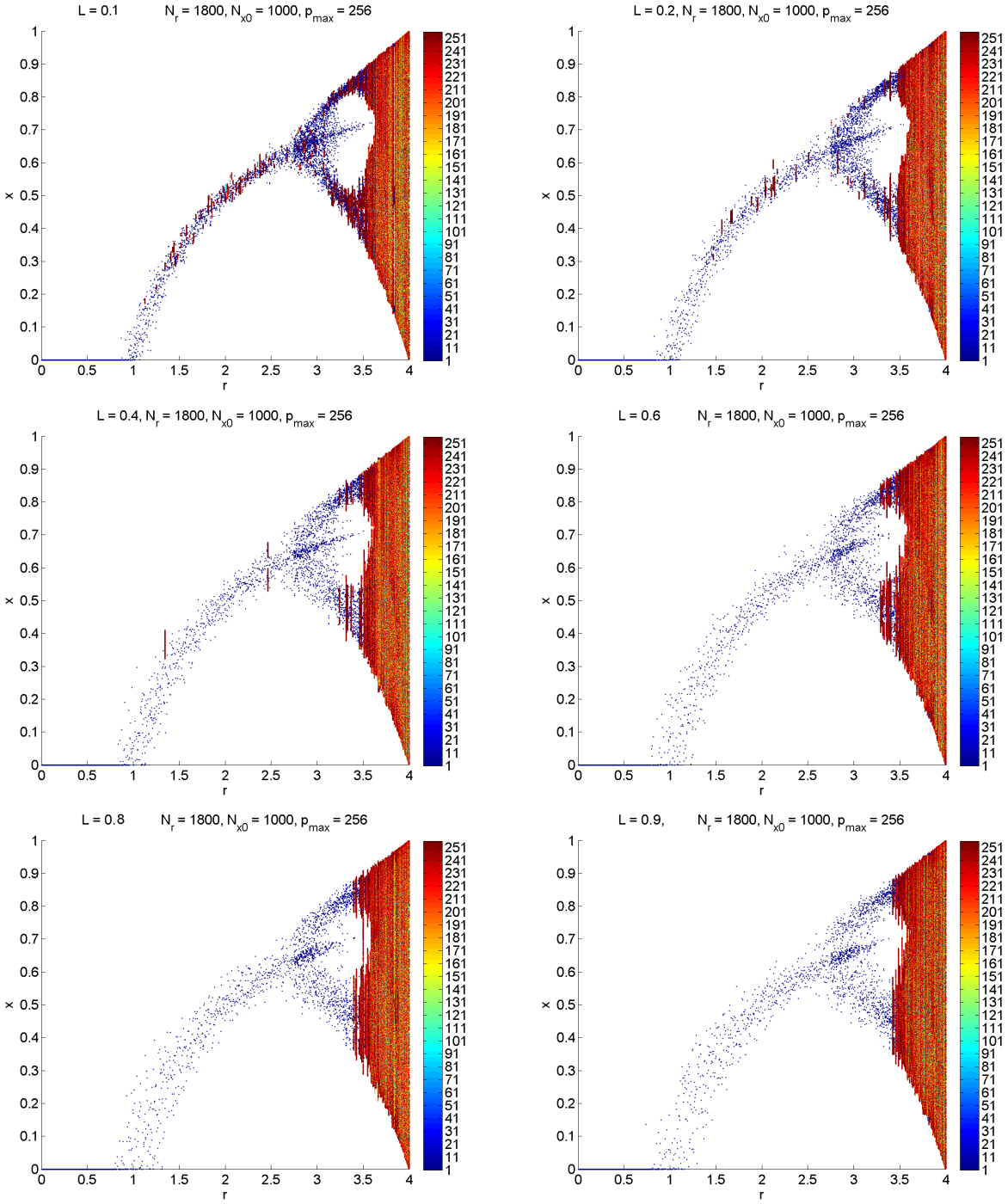


Figure 3.3: Bifurcation diagram of the random logistic map, where $r \in [0, 4]$, $\Delta r = 0.002$, $N = 100$, the number of initial conditions tested is N_{x_0}, $L \in \{0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$, and $\sigma = \frac{1}{2}\sigma_{max}$. Plots are read left to right, and top to bottom. Number of simulations is 1.8 million. The legend shows the color for each period. Orbits up to period 256 were checked.



Examining the Lyapunov exponent of the random logistic map for the case where σ is large gives some insight on whether the map is possibly chaotic. The exponent of the deterministic map is on the top left of Figure 3.4, and we see there is a point in r where stable behavior transitions to chaotic behavior (just under $r = 3.6$). This point is referred to as the Feigenbaum period-doubling accumulation point. In the randomized case, the delimitation is unclear. However one feature of the deterministic diagram seems to be preserved, and that is the negative spike around $r = 3.8$, which corresponds to a window of stability in the deterministic map (Table 1.2).

Figure 3.5 demonstrates results from simulating the random logistic map where the standard deviation σ of $\xi(x)$ is chosen to be half of the maximal value from (2.15). The positive Lyapunov exponents of the randomized logistic map for the case where σ is small in Figure 3.5 imply that even a halved standard deviation is enough to incur chaotic behavior. The negative spike around $r = 3.8$, which corresponds to an island of stability in the deterministic map, is more prominent when σ is small than when large. Between Figure 3.4 and Figure 3.5, it appears that decreasing σ causes the plot of Lyapunov exponents of the random map to fall more in line with the plot for the deterministic map. There are fewer positive exponents in Figure 3.5 than in Figure 3.4, and remnants of other islands of stability from the deterministic map appear in Figure 3.5 (e.g. $r \approx 3.95$).

The Lyapunov exponents for the random map were calculated according to the formula in (1.1) with $N_\lambda = n = 10,000$. For the logistic map,

$$\begin{aligned}
\lambda(x_0) &= \frac{1}{n}\sum_{i=0}^{n-1} \ln|f'(x_i)| \\
&= \frac{1}{10000}\sum_{i=0}^{9999} \ln|R'(x_i)x_i(1-x_i) + R(x_i)(1-2x_i)| \\
&= \frac{1}{10000}\sum_{i=0}^{9999} \ln|e^{\xi(x_i)}\xi'(x_i)x_i(1-x_i) + e^{\xi(x_i)}(1-2x_i)|.
\end{aligned} \tag{3.1}$$

Values of r were chosen in $[3, 4]$ with step size $\Delta r = 0.001$. The initial condition x_0 was fixed for all simulations at $x_0 = 0.7$.

A histogram describing the average fraction of observed period p orbits is shown in Figure 3.6

Figure 3.4: The Lyapunov exponent for the deterministic logistic map (top left) is compared to the Lyapunov exponent of the random logistic map for $L \in \{0.05, 0.1, 0.5, 0.7, 0.9\}$, where $x_0 = 0.7$ for $r \in [3, 4]$, and $\sigma = \sigma_{max}$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.

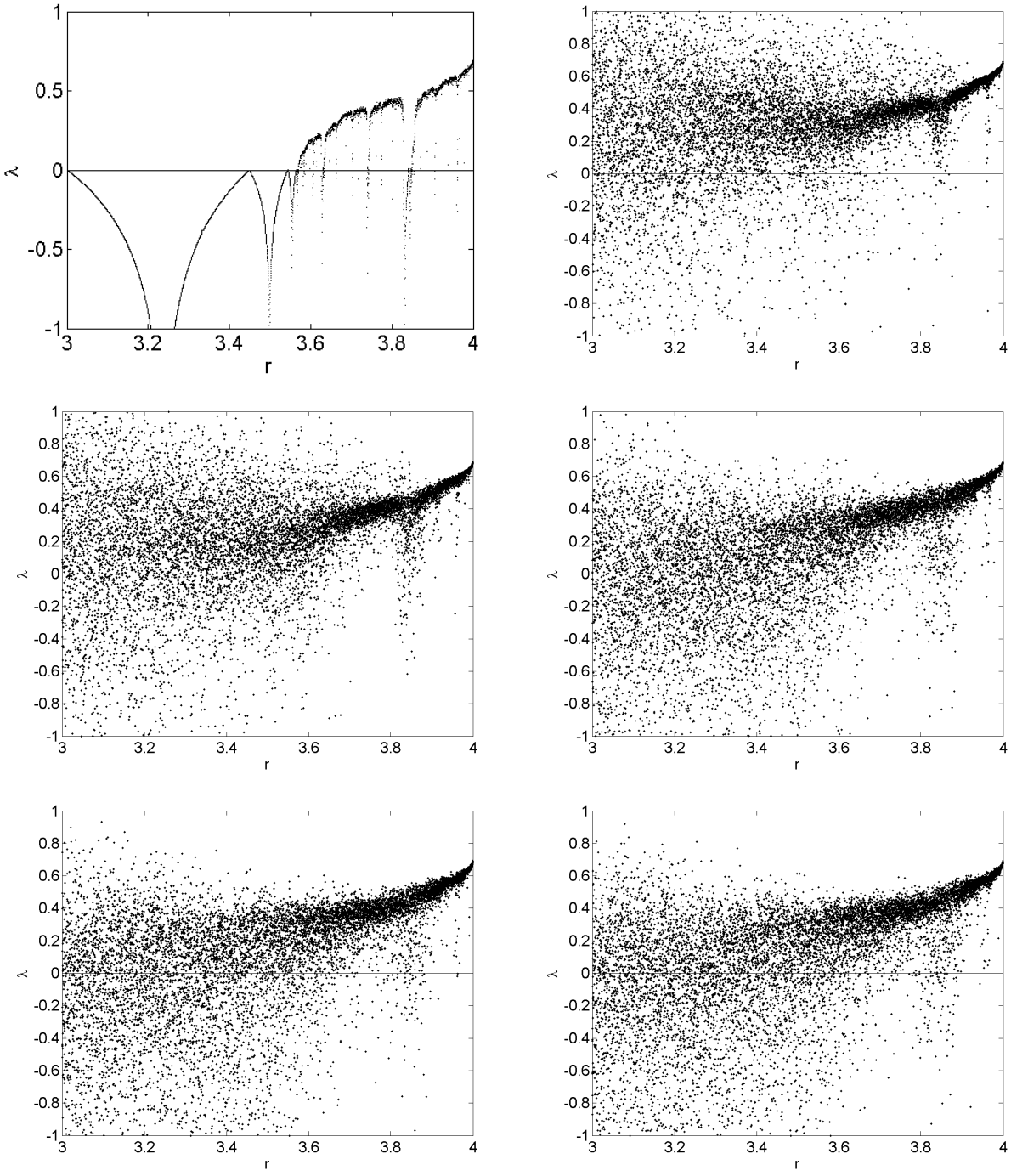
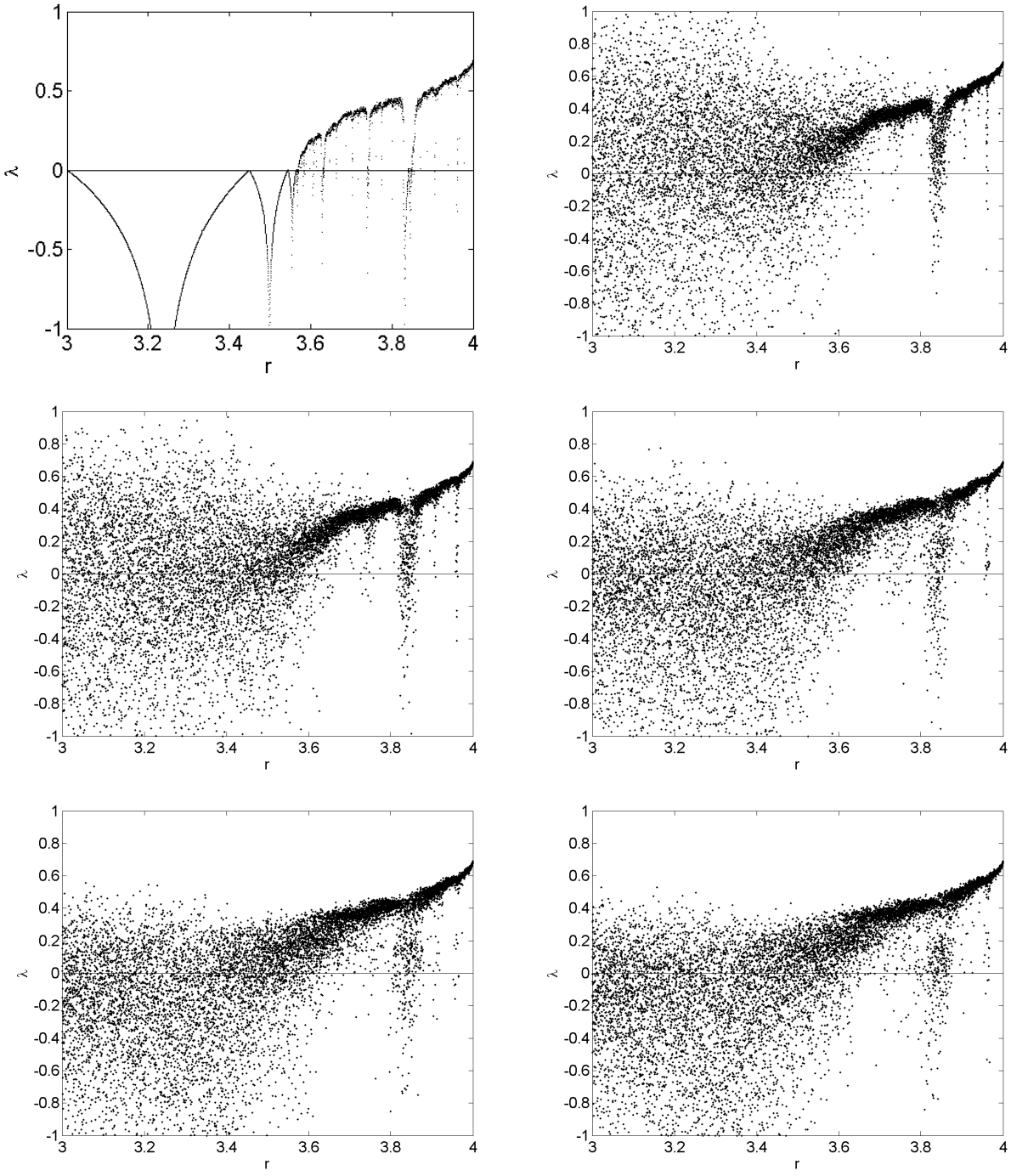


Figure 3.5: The Lyapunov exponent for the deterministic logistic map (top left) is compared to the Lyapunov exponent of the random logistic map for $L \in \{0.05, 0.1, 0.4, 0.8, 0.9\}$, where $x_0 = 0.7$ for $r \in [3, 4]$, and $\sigma = \frac{1}{2}\sigma_{max}$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.



and 3.8. The histogram in Figure 3.6 implies that for $r = 3.3$, the most commonly observed stable orbit is period 2, and the distribution of periodic orbits seems to decay exponentially, however a log-scale plot of the histogram shows the distribution may not be exponential (Figure 3.7). This figure corresponds with the newly stabilized region of the logistic map. Figure 3.8 scans the distribution of period for a smaller value of $r = 1.2$. Here, the most dominant period is 1, and the decaying frequency of high-period orbits seems more pronounced. The distribution of period for any given (r, L) pair offers insight on the type of orbits that the random process solicits.

The histograms were generated by testing 10 random initial conditions x_0 500 times each, resulting in 5,000 total simulations. Each time any given x_0 was tested, a new set of random Fourier modes $\hat{\xi}_n$ were drawn from either a uniform distribution. After iterating the random map 1,000 times for some r and this set of Fourier modes, an orbit was denoted period p if $x_p = x_{n+p}$ within a tolerance of $\epsilon = 10^{-6}$. Periodicity was checked up to $p_{max} = 100$. The histograms count unique periodic orbits, so if two or more initial conditions under the same random process converged to the same periodic orbit, only one was counted.

Figure 3.6: Average fraction of period p orbits for the random logistic map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $r = 3.3$ and $\sigma = \sigma_{max}$. For (L, N, σ), we have $(0.025, 400, 0.0087808)$ (left), $(0.05, 200, 0.012418)$ (middle), and $(0.1, 100, 0.017565)$ (right).

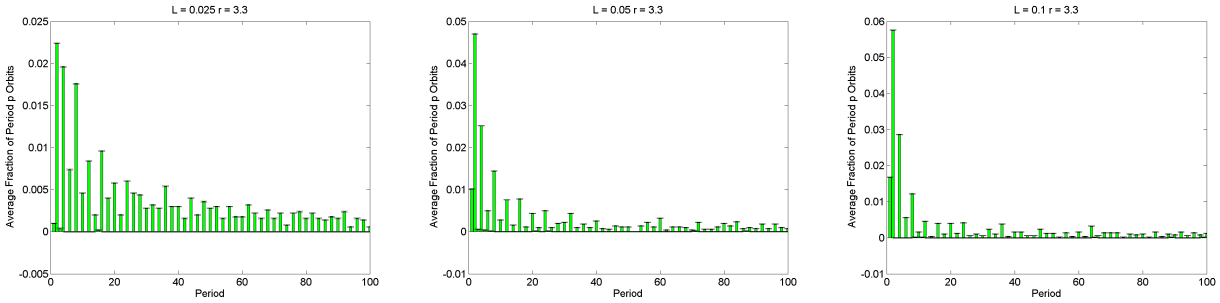


Figure 3.7: A log-scale plot of the average fraction of period p orbits for the random logistic map, where 5000 simulations are plotted. $r = 1.2, L = 0.1, N = 100$, and $\sigma = \sigma_{max}$.

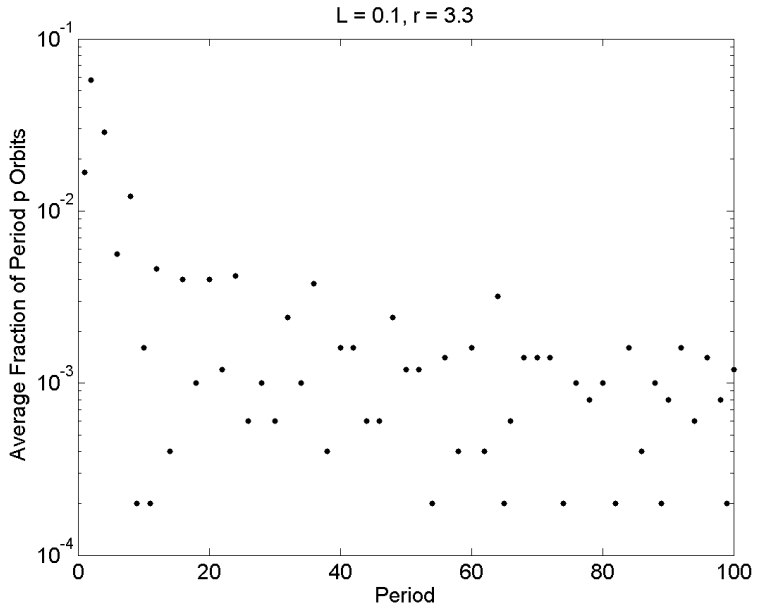
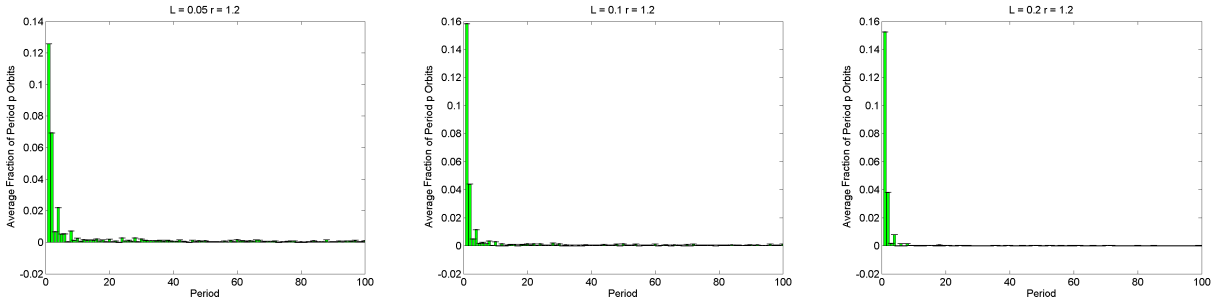


Figure 3.8: Average fraction of period p orbits for the random logistic map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $r = 1.2$ and $\sigma = \sigma_{max}$. For (L, N, σ), we have $(0.05, 200, 0.07772)$ (left), $(0.1, 100, 0.10993)$ (middle), and $(0.2, 100, 0.15556)$ (right).



For the case where $\sigma = \frac{1}{2}\sigma_{max}$, a histogram describing the average fraction of observed period p orbits is shown in Figure 3.9 and 3.10. These histograms confirm the observation that reducing σ causes the map to resemble the deterministic map more closely, which is also shown in Figure 3.3,

the bifurcation diagrams. The frequency of high-period orbits is much lower in Figure 3.9 and 3.10 than in Figure 3.6 and 3.8, where σ is twice as large. Nonetheless, the general trend in Figure 3.9 and 3.10 implies the distribution of period is similar for both values of σ.

Figure 3.9: Average fraction of period p orbits for the random logistic map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $r = 3.22$ and $\sigma = \frac{1}{2}\sigma_{max}$. For (L, N, σ), we have $(0.025, 400, 0.0049504)$ (left), $(0.05, 200, 0.0070012)$ (middle), and $(0.1, 100, 0.0099027)$ (right).

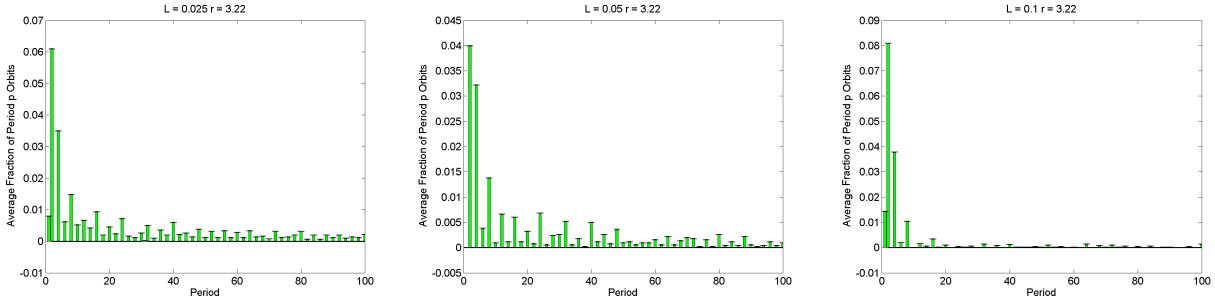
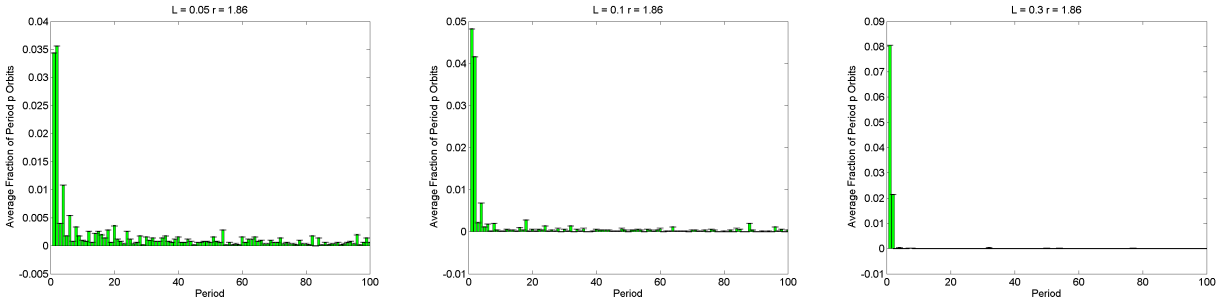


Figure 3.10: Average fraction of period p orbits for the random logistic map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $r = 1.86$ and $\sigma = \frac{1}{2}\sigma_{max}$. For (L, N, σ), we have $(0.05, 200, 0.024715)$ (left), $(0.1, 100, 0.034957)$ (middle), and $(0.3, 100, 0.060648)$ (right).



3.2 Implementation of Randomness in the Circle Map

We explored how the random circle map for uniformly distributed $\hat{\xi}_n$ and normally distributed $\hat{\xi}_n$ changes in terms of its Arnold tongues, devil's staircases, Lyapunov exponents, and periodic orbit distribution. We examined the distribution of rotation numbers using a kernel density estimator. Histograms of the observed frequency of a period p orbit attempt to describe the period distribution in the random circle map. The parameter α that scales the variance of the Fourier modes of the spatial process $\xi(x)$ from (2.8) was tested for two values: $\alpha = 10^{-5}$ and $\alpha = \frac{1}{2}10^{-5}$. Essentially, this parameter controls the magnitudes of the randomly drawn $\hat{\xi}_n$. The parameter α also scales the covariance of $\xi(x)$, shown in (2.9). These two values were explored in the case where $\hat{\xi}_n$ was uniformly and normally distributed.

In the Arnold tongue diagrams (Figure 3.11, 3.12, 3.24, 3.26), each value of $k \in [0, 1.5]$ and $\omega \in [0, 1]$ was tested according to the discretization of $\Delta k = 0.0015, \Delta\omega = 0.001$ and initial condition $x_0 = 0.7$. If iterating the map for any given set of parameters resulted in finding no periodic orbit of period p_{max} or less, then the pixel corresponding to this value of (ω, k) was colored black. Otherwise, the pixel for this (ω, k) pair was colored according to the orbit period. An orbit was denoted period p if, after iterating the map 1,000 times, $x_p = x_{n+p}$ within a tolerance of $\epsilon = 10^{-6}$.

Plotting the change in rotation number ρ as ω is varied over $[0, 1]$ results in a devil's staircase (Figure 3.21, 3.22, 3.34, and 3.35). The figures relating to the devil's staircase fixed a value of L and k over 10,000 values of $\omega \in [0, 1]$. Each L tested had the same set of Fourier modes $\hat{\xi}_n$ for all 10,000 ω values. The lift of the circle map was used to find the rotation number, and it was assumed that iterating the lift 1,000 times was enough to converge on a rotation number, if one existed.

The Lyapunov exponents for the random map were calculated according to the formula

in (1.1) with $N_\lambda = n = 10,000$ (Figure 3.13, 3.14, 3.27, 3.28, and 3.29). For the circle map,

$$\begin{aligned}\lambda(x_0) &= \frac{1}{n}\sum_{i=0}^{n-1}\ln|f'(x_i)| \\ &= \frac{1}{10000}\sum_{i=0}^{9999}\ln|1+\Omega'(x_i)-k\cos(2\pi x_i)| \\ &= \frac{1}{10000}\sum_{i=0}^{9999}\ln|1+e^{\xi(x_i)}\xi'(x_i)-k\cos(2\pi x_i)|.\end{aligned} \tag{3.2}$$

Values of ω were chosen in $[0,1]$ with step size $\Delta\omega = 0.001$ for the case where k was fixed, and values of k were chosen in $[0,5]$ with step size $\Delta k = 0.005$ for the case where ω was fixed. The initial condition x_0 was the same for all simulations: $x_0 = 0.7$.

To produce the histograms, 10 random initial conditions x_0 were tested 500 times each, resulting in 5,000 total simulations (Figure 3.15, 3.16, 3.18, 3.19, 3.30, 3.31, 3.32, and 3.33). Each time any given x_0 was tested, a new set of random Fourier modes $\hat{\xi}_n$ were drawn from either a uniform or normal distribution. After iterating the random map 1,000 times for some ω, k pair and this set of random variables, an orbit was denoted period p if $x_p = x_{n+p}$ within a tolerance of $\epsilon = 10^{-6}$. Periodicity was checked up to $p_{max} = 100$. The histograms count unique periodic orbits, so if two or more initial conditions converged to the same periodic orbit, only one was counted.

3.2.1 Uniform Distribution

The randomized circle map for uniformly distributed $\hat{\xi}_n$ has a set of Arnold tongues (Figure 3.11 and 3.12) that has almost no similarity to the deterministic case. For low values of L, we have lost the shape of the tongues, and the diagram becomes asymmetrical. For larger values of L, the distinctive tongues and the overall symmetry is recovered. The randomness appears to have an overall destabilizing effect on the dynamics of the map. For $L = 0.05$, we also observe the presence of high-period orbits in the region where there is typically only period 1 fixed points (upper left). As L increases, this region morphs into predominantly stable period 1 orbits.

For $L = 0.05$, there is a noticeable lack of high-period orbits for $\omega < 0.2$ when $\alpha = \frac{1}{2}10^{-5}$ (Figure 3.12) compared to the case where $\alpha = 10^{-5}$ (Figure 3.11). In addition to this observation,

when $L = 0.3$, there is a region of stable period 2 orbits for $\omega > 0.9$ when $\alpha = \frac{1}{2}10^{-5}$, yet this region is absent for $\alpha = 10^{-5}$. Also, the stable period 4 region for $\omega \approx 0.5$ is much smaller when α is half as large. Increasing α seems to increase the number of high-period orbits for certain values of $\omega < 0.2$, eliminate period 2 orbits when $\omega > 0.9$, and reduce the number of period 4 orbits in the center tongue.

Figure 3.13 is a plot of the Lyapunov exponents for a fixed k and varying ω, whereas Figure 3.14 fixes ω and varies k. A comparison of the Lyapunov exponent of the deterministic and random case (Figure 3.13 and Figure 3.14) partially confirms the idea that the noise is destabilizing; nearly no features of the deterministic graph are preserved (for small L), and the high density of positive values indicates chaotic behavior. Moreover, Figure 3.13 demonstrates that there is a skewed distribution of Lyapunov exponents on the right side of the graph for all values of L, compared to the left side. However, this trend is not observed in Figure 3.14.

Figure 3.11: The Arnold tongues for $k \in [0, 1.5]$, $\Delta k = 0.0015$, $\omega \in [0, 1]$, $\Delta\omega = 0.001$, $\alpha = 10^{-5}$, $\hat{\xi}_n \sim Unif(-M_n, M_n)$ and $L_j \in \{0.05, 0.1, 0.3, 0.5, 0.7, 0.9\}$. Δk and $\Delta\omega$ represent the step size in the discretization of k on [0,1.5] and ω on [0,1]. Plots are ordered left to right, and top to bottom. The legend to the right demonstrates the period and corresponding color.

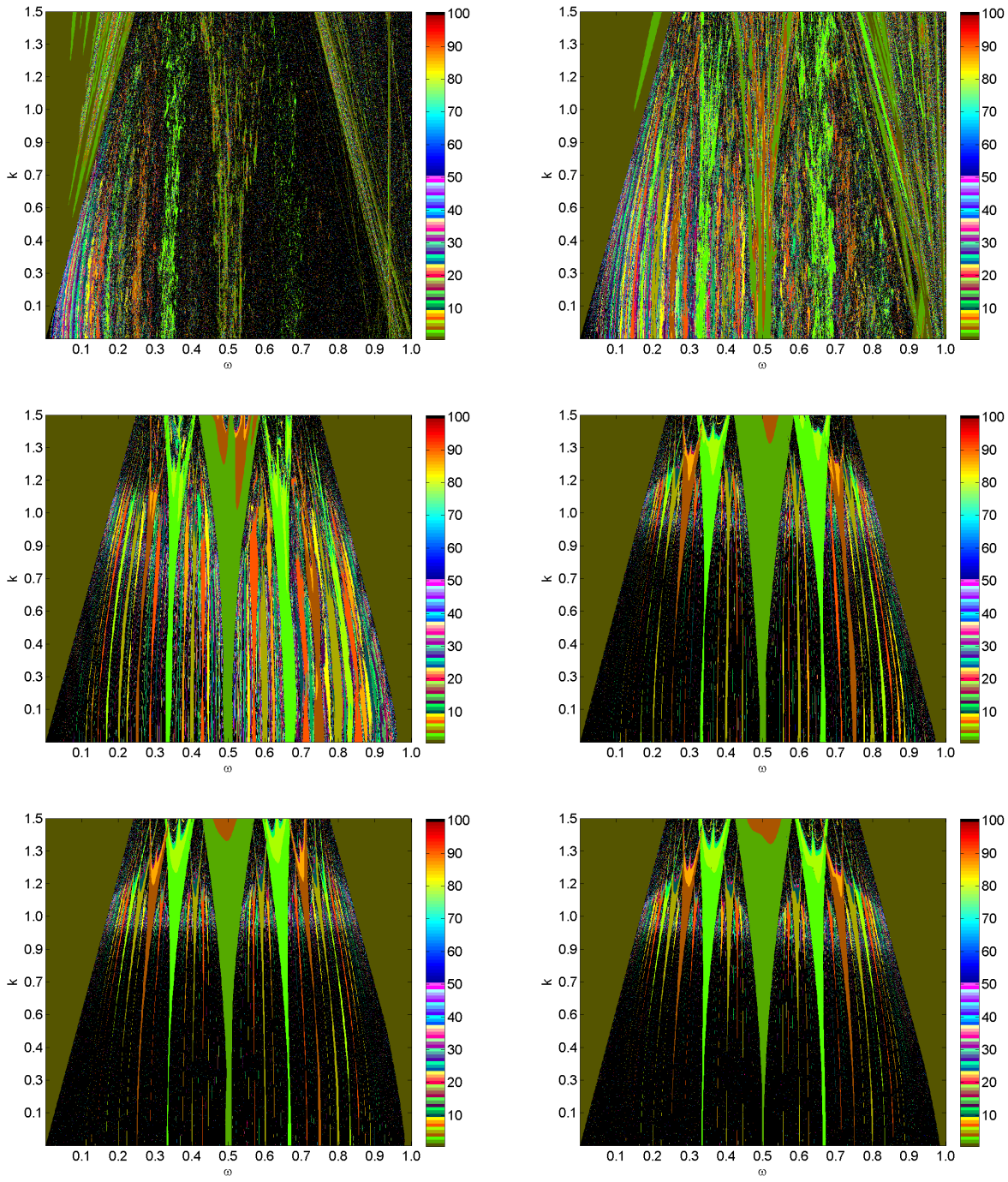


Figure 3.12: The Arnold tongues for $k \in [0, 1.5]$, $\Delta k = 0.0015$, $\omega \in [0, 1]$, $\Delta\omega = 0.001$, $\alpha = \frac{1}{2}10^{-5}$, $\hat{\xi}_n \sim Unif(-M_n, M_n)$ and $L_j \in \{0.025, 0.05, 0.1, 0.3, 0.5, 0.9\}$. Δk and $\Delta\omega$ represent the step size in the discretization of k on [0,1.5] and ω on [0,1]. Plots are ordered left to right, and top to bottom. The legend to the right demonstrates the period and corresponding color.

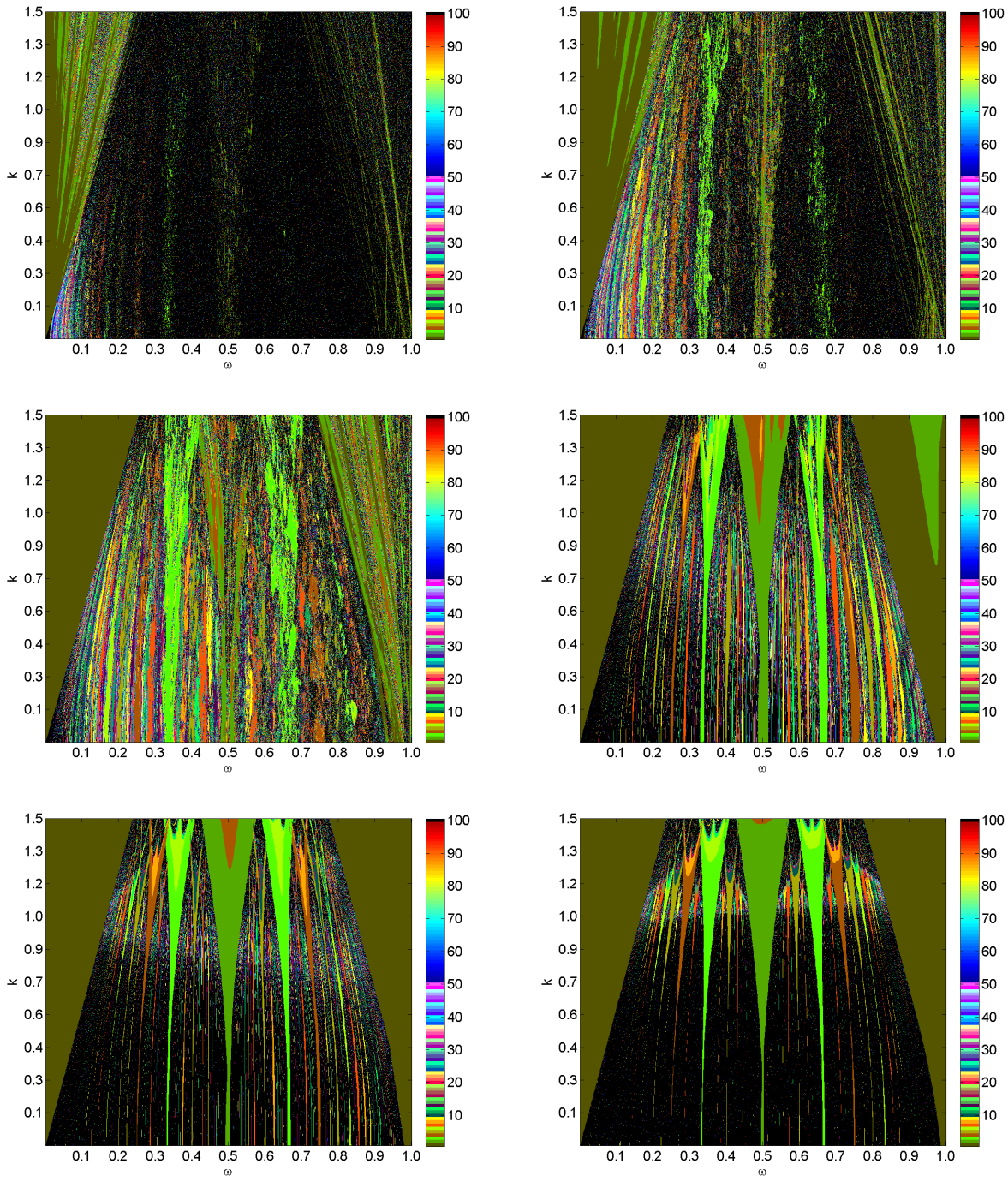


Figure 3.13: The Lyapunov exponent for the deterministic circle map (top left) is compared to the Lyapunov exponent of the random circle map for $L \in \{0.05, 0.1, 0.3, 0.5, 0.9\}$, where $x_0 = 0.7$, $k = 2$, $\alpha = 10^{-5}$, and $\hat{\xi}_n \sim Unif(-M_n, M_n)$ for $\omega \in [0, 1]$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.

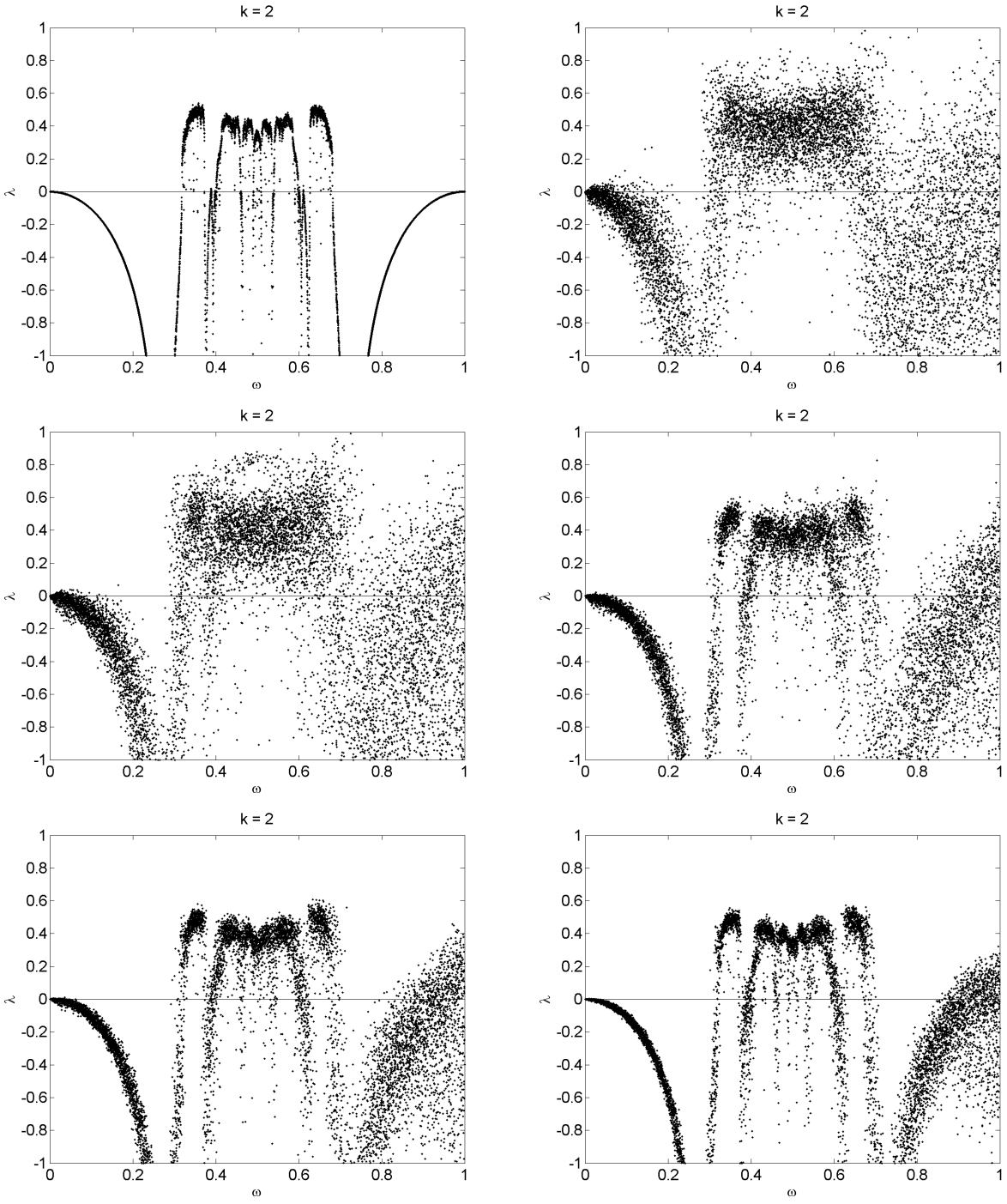
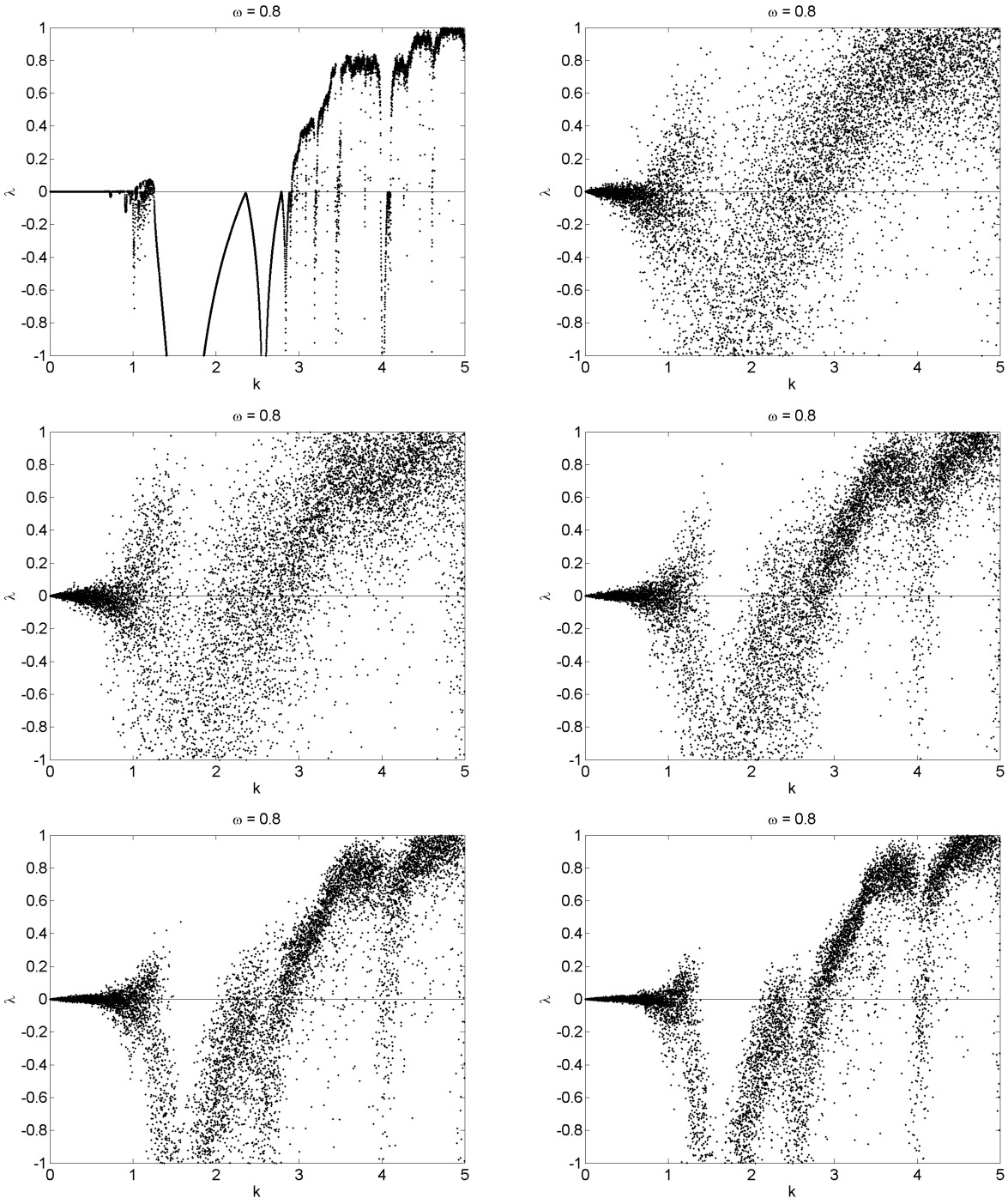


Figure 3.14: The Lyapunov exponent for the deterministic circle map (top left) is compared to the Lyapunov exponent of the random circle map for $L \in \{0.05, 0.1, 0.3, 0.5, 0.7\}$, where $x_0 = 0.7$, $\omega = 0.8$,$\alpha = 10^{-5}$, and $\hat{\xi}_n \sim Unif(-M_n, M_n)$ for $k \in [0, 5]$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.



The histograms in Figure 3.15 and 3.16 demonstrate results from calculating the average fraction of unique period p orbits over 5,000 simulations of the random circle map where $\hat{\xi}_n \sim Unif(-M_n, M_n)$. In Figure 3.15, the two parameters are fixed at $\omega = 0.9$ and $k = 1$, and in Figure 3.16, $\omega = 0.9$ and $k = 1.5$.

Figure 3.15: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = 10^{-5}$, $\omega = 0.9$ and $k = 1$. For (L, N), we have $(0.025, 400)$ (left), $(0.05, 200)$ (middle), and $(0.1, 100)$ (right).

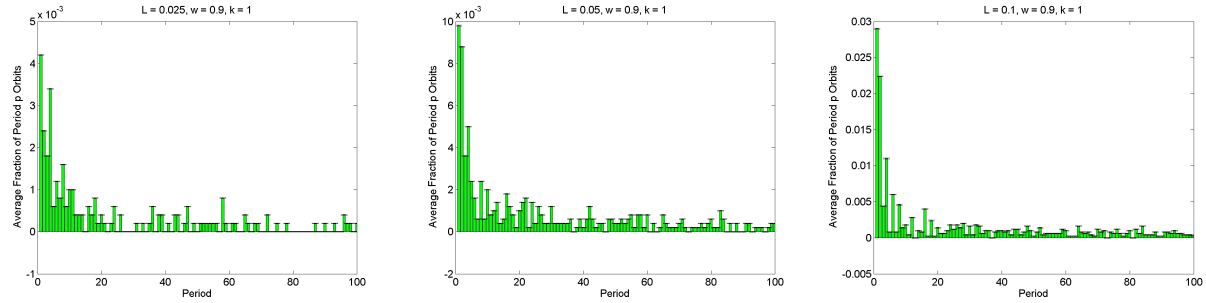
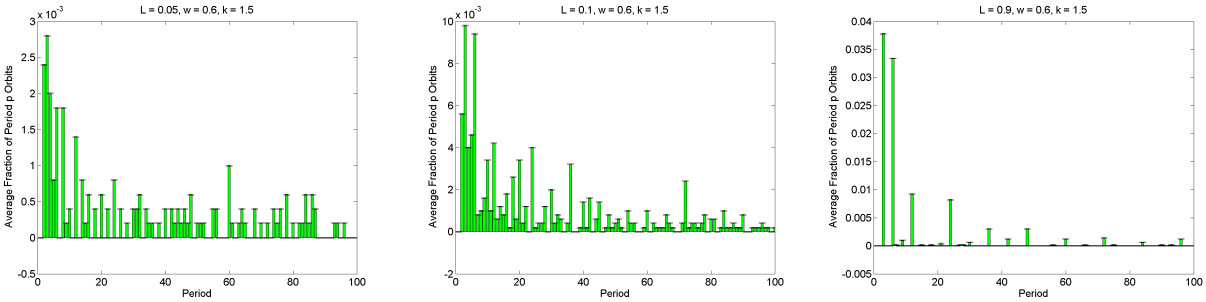


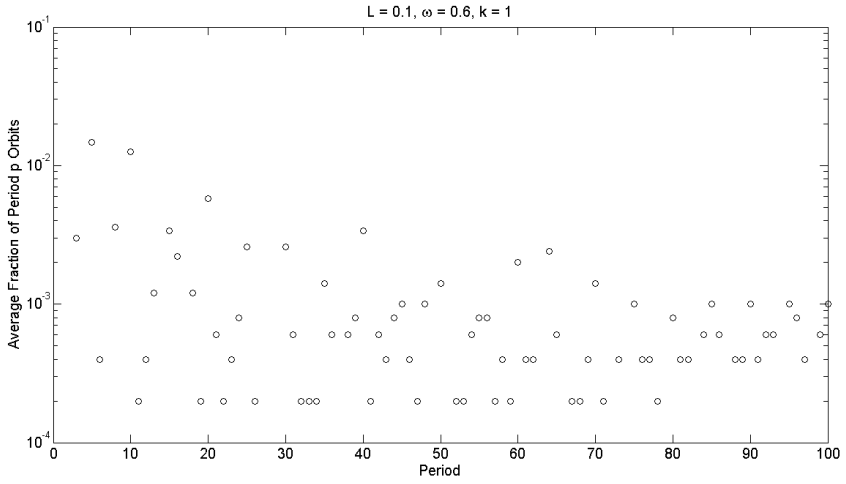
Figure 3.16: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = 10^{-5}$, $\omega = 0.6$ and $k = 1.5$. For (L, N), we have $(0.05, 200)$ (left), $(0.1, 100)$ (middle), and $(0.9, 100)$ (right).



The histograms in the previous two figures (3.15, 3.16) seemed to obscure some results due to the linear scale of the y-axis. A plot of the average fraction of orbits (Figure 3.17) uses a

logarithmic scale. It is clear that the largest number of orbits over 5,000 simulations is period 5. The parameters tested in this simulation were for $k = 1, \omega = 0.6$. Interestingly, the semilog-plot demonstrates the distribution of periodic orbits is not exponential, despite the sharp decline in frequency, since the shape of the graph is not linear.

Figure 3.17: Log-scale plot of average fraction of period p orbits for the random circle map, where $L = 0.1$, $\omega = 0.6$, $\alpha = 10^{-5}$, $\hat{\xi}_n \sim Unif(-M_n, M_n)$ and $k = 1$. Results from 5,000 simulations of these parameters are plotted, using a logarithmic scale for the y-axis.



Halving α has a big impact on the histograms. The histograms in Figure 3.15 and 3.16 are qualitatively different from the ones in Figure 3.18 and 3.19. First, the histograms for $L = 0.025, \omega = 0.9, k = 1$ (left side) of Figure 3.15 and 3.18 demonstrate that the observed frequency of period 1 and 2 orbits have opposite trends when α is halved. For $\alpha = \frac{1}{2}10^{-5}$, the mean number of observations of period 1 orbits $\bar{p}_1 = 0.0028$ and $\bar{p}_2 = 0.0048$, but $\bar{p}_1 = 0.0042$ and $\bar{p}_2 = 0.0024$ when α is twice as large. In other words, the reverse is true. Also notably, the histograms for $L = 0.9, \omega = 0.6, k = 1.5$ between Figure 3.16 and 3.19 (right side) are quite different, although the only parameter changed was α. In Figure 3.16, period 3 and 6 orbits dominate the plot evenly, with a mean $\bar{p}_3 = 0.0378$ and $\bar{p}_6 = 0.0334$. Yet, in Figure 3.19, $\bar{p}_3 = 0.0284$ and $\bar{p}_6 = 0.0572$; period 3 is half as frequently observed than period 6. Other than these two observations, the histograms

are quite similar. Unlike the random logistic map, it does not seem that the distribution of period is consistent in the circle map when the variance of the Fourier modes is reduced.

Figure 3.18: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = \frac{1}{2}10^{-5}$, $\omega = 0.9$ and $k = 1$. For (L, N), we have $(0.025, 400)$ (left), $(0.05, 200)$ (middle), and $(0.1, 100)$ (right).

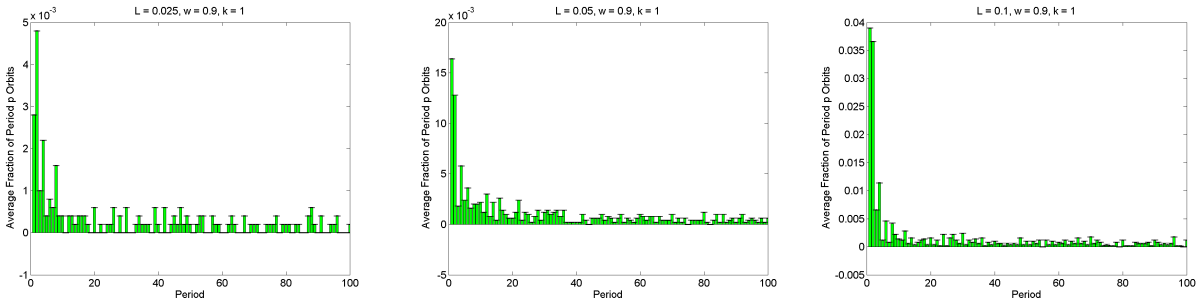
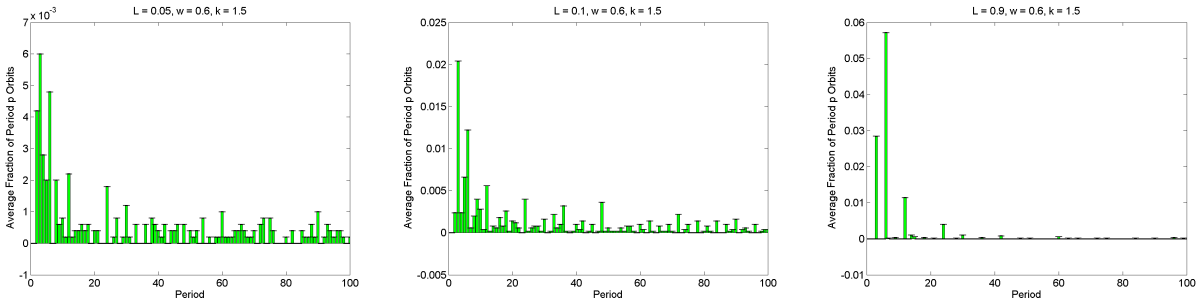


Figure 3.19: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = \frac{1}{2}10^{-5}$, $\omega = 0.6$ and $k = 1.5$. For (L, N), we have $(0.05, 200)$ (left), $(0.1, 100)$ (middle), and $(0.9, 100)$ (right).



A log-plot of the average fraction of orbits (Figure 3.20) uses a logarithmic scale, like in Figure 3.17. It is clear that the largest number of orbits over 5,000 simulations is period 5 and there are many stable high-period orbits for both the case where $\alpha = 10^{-5}$ and $\alpha = \frac{1}{2}10^{-5}$. The parameters tested in this simulation were for $k = 1, \omega = 0.6$. Interestingly, both the semilog-plots

demonstrate a non-exponential distribution of period, despite the sharp drop off in frequency. One difference between the plots is that for $\alpha = \frac{1}{2}10^{-5}$, the range of average fraction of periodic orbits seems wider than for $\alpha = 10^{-5}$.

Figure 3.20: Average fraction of period p orbits for the random circle map, where $L = 0.1$, $\omega = 0.6$, $\alpha = \frac{1}{2}10^{-5}$, $\hat{\xi}_n \sim Unif(-M_n, M_n)$ and $k = 1$. Results from 5,000 simulations of these parameters are plotted, using a logarithmic scale for the y-axis.

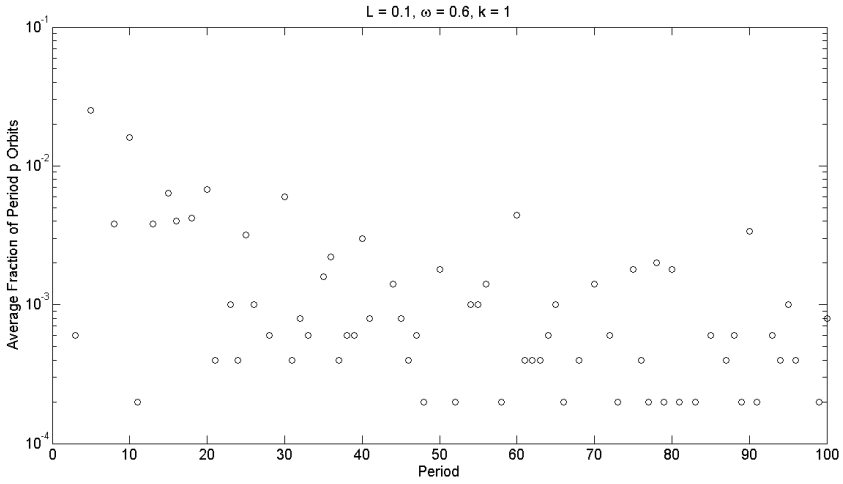


Figure 3.21 and Figure 3.22 are plots of the rotation number of the random circle map. Recall that a rational rotation number $\rho = p/q$ indicates that there is a period q orbit in the map, whereas an irrational ρ indicates quasiperiodic orbits, which means chaos is a possibility. Figure 3.21 implies that decreasing L introduces the most noise in the plot of rotation numbers, and Figure 3.22 shows that for $L = 0.1$, there may be chaos in the random map for values of $k < 1$. In the deterministic map, chaos appears only for $k > 1$.

Figure 3.21: The devil's staircase for $L = 0.05$ (left), $L = 0.3$ (middle), and $L = 0.5$ (right), where $k = 1$,$\alpha = 10^{-5}$ and $\hat{\xi}_n \sim Unif(-M_n, M_n)$. For small L, the noise is more pronounced than for large L.

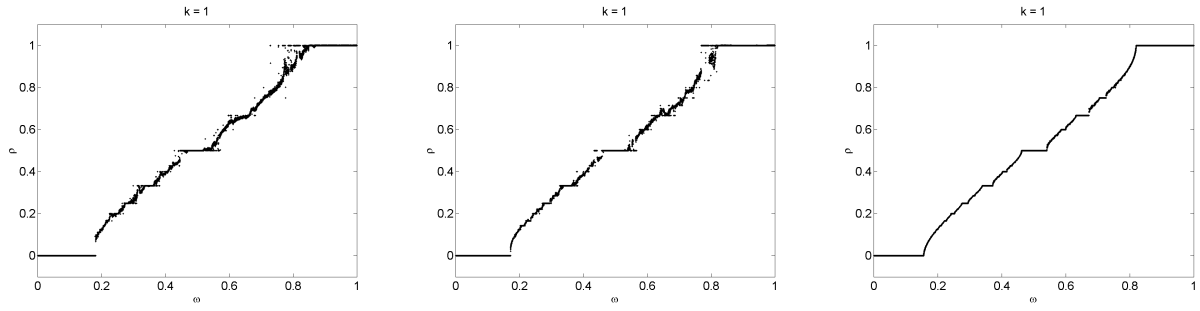


Figure 3.22: The devil's staircase for $k = 0.7$ (left), $k = 1$ (middle), and $k = 1.5$ (right). In each plot, $\alpha = 10^{-5}$,$\hat{\xi}_n \sim Unif(-M_n, M_n)$ and $L = 0.1$.

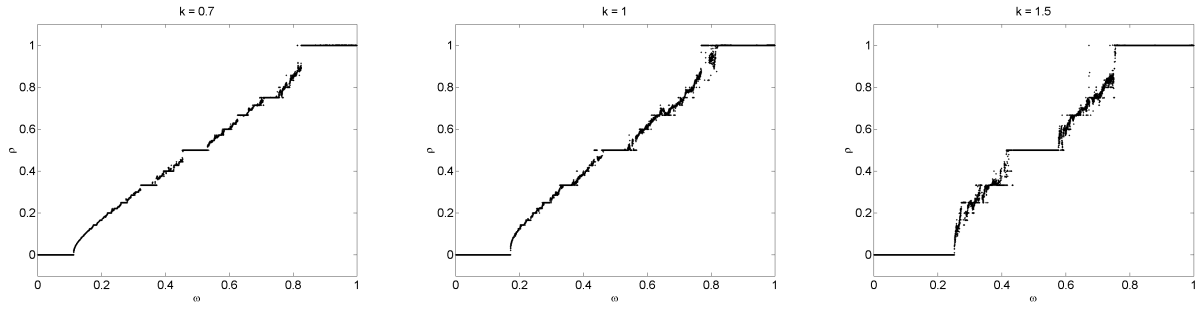
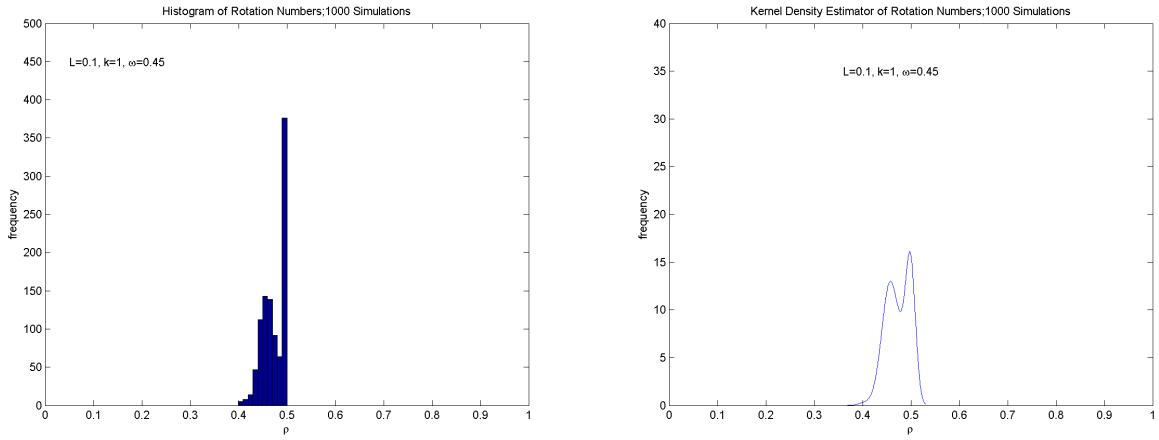


Figure 3.23 is a histogram of the observed number of rotation numbers for fixed $L = 0.1, k = 1, \alpha = 10^{-5}, \omega = 0.45$. Next to the histogram is the kernel density estimator for this distribution of rotation numbers. Kernel density estimation is a non-parametric way of estimating the probability density function of a random variable. This method makes no assumptions regarding the density functions of the observed random variables. The key parameter in kernel density estimation is the bandwidth, which is also known as a smoothing parameter. The smaller the bandwidth, the less smooth the density estimate. This may exaggerate some characteristics of the sample. In MATLAB, the default bandwidth is $h = 1.5141$, and this bandwidth is reflected in Figure 3.23. The kernel density estimation suggests that for this (L, k, ω)-tuple, the distribution of rotation numbers may be normal. Using a larger h would smooth the estimation so that the curve resembles the normal density function. An alternative way to see a normal shape of the data is to exclude

the data in the histogram for $\rho = 0.5$. In the deterministic map, we would see only $\rho = 0.5$ since $\omega = 0.45$ corresponds with period 2 orbits. Excluding the data for $\rho = 0.5$ would give a sense of how many rotation numbers are a function of the spatially random process.

Figure 3.23: Histogram of rotation numbers in the random circle map, where $L = 0.1$, $k = 1$, $\alpha = 10^{-5}$, $\hat{\xi}_n \sim Unif(-M_n, M_n)$, and $\omega = 0.45$. Results from 1000 simulations are plotted.



3.2.2 Normal Distribution

Like the uniformly distributed case, the randomized circle map for normally distributed $\hat{\xi}_n$ has a set of Arnold tongues that has almost no similarity to the deterministic case (Figure 3.24, 3.25, 3.26). The same trend in L is observed: small L values seem to correspond with asymmetry and loss of tongue-shape, and aspects of the deterministic map are recovered for larger L. The magnitude of the constant scaling σ was reduced to $\alpha = \frac{1}{2}10^{-5}$ in Figure 3.26. Generally, the two cases where different values of α were tested resulted in similar figures. The reduced noise in Figure 3.26 is reflected especially when $L = 0.1$; the high-period orbits for $\omega < 0.2, k > 0.9$ from Figure 3.24 are absent. It seems like there is a big qualitative change between $L = 0.1$ and $L = 0.3$, so Figure 3.25 shows the evolution of the bifurcation diagram as L is increased by a factor of $\Delta L = 0.025$ from [0.125,0.25]. The smaller step size in L sheds light on how the tongues transform from their initial shape to a diagram that loosely resembles the deterministic map. First,

the right side of the diagram transforms from black (period $p \geq 100$ or no stable orbit at all) for $L = 0.125$ to having bands of period 1 and 2 interspersed with high-period orbits for $L = 0.15$. Second, the region switches back to being predominantly black for $L = 0.175$. Another shift in L to $L = 0.2$ results in wider period 1 and 2 bands. Next, for $L = 0.225$, the diagram fluctuates back to a seemingly chaotic state, although there is a wide period 1 band of orbits present as well. Finally, for $L = 0.25$, there is very little left of the high-period orbits, and the main colors match period 1 and 2.

Again, the randomness appears to have an overall destabilizing effect on the dynamics of the map in both the uniform and normal tongue diagrams. In comparison to the uniform case, for $L = 0.05$, there seem to be more high-period orbits in the region where there is typically only period 1 fixed points (upper left). Furthermore, the shape of the tongues emerges more clearly for $L = 0.1$ in the uniform case, and the normal case appears more disjoint. Perhaps the difference between the normal and uniform cases is best highlighted in the diagrams for $L = 0.3$. There are many high-period orbits for $\omega \approx 0.5$ in the uniform case, but very few in the normal case. Also, there is a period 2 tongue on the right side of the plot in the normal case, which is absent in the uniform case. For $L = 0.9$, the uniform case exhibits period 2 and 4 orbits for $\omega = 0.5$, but the normal case only shows period 2 orbits, hinting that low-period orbits are more stable for the normal case.

The Lyapunov exponents for a fixed k and varying ω (Figure 3.27) and exponents for fixed ω and varying k (Figure 3.28) possess some similar qualities to the uniform case. The exponents partially confirm the idea that the noise is destabilizing, as few features of the deterministic graph are preserved. The high density of positive values point to chaotic behavior. The skewed distribution of Lyapunov exponents on the right side of the graph from Figure 3.13 is replicated in Figure 3.27. In both the uniform and normal case (Figure 3.14 and Figure 3.28), as L grows larger, more features of the deterministic plot become apparent. For instance, when $L = 0.05$, one can only make out a very general outline of the overall shape of the Lyapunov exponents, but for $L = 0.9$ and $L = 0.7$, there is a tighter bound on how far the exponents stray from their deterministic configuration.

Figure 3.24: The Arnold tongues for $k \in [0, 1.5]$, $\Delta k = 0.0015$, $\omega \in [0, 1]$, $\Delta\omega = 0.001$, $\alpha = 10^{-5}$, $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ and $L_j \in \{0.025, 0.05, 0.1, 0.3, 0.5, 0.9\}$. Δk and $\Delta\omega$ represent the step size in the discretization of k on [0,1.5] and ω on [0,1]. Plots are ordered left to right, and top to bottom. The legend to the right demonstrates the period and corresponding color.

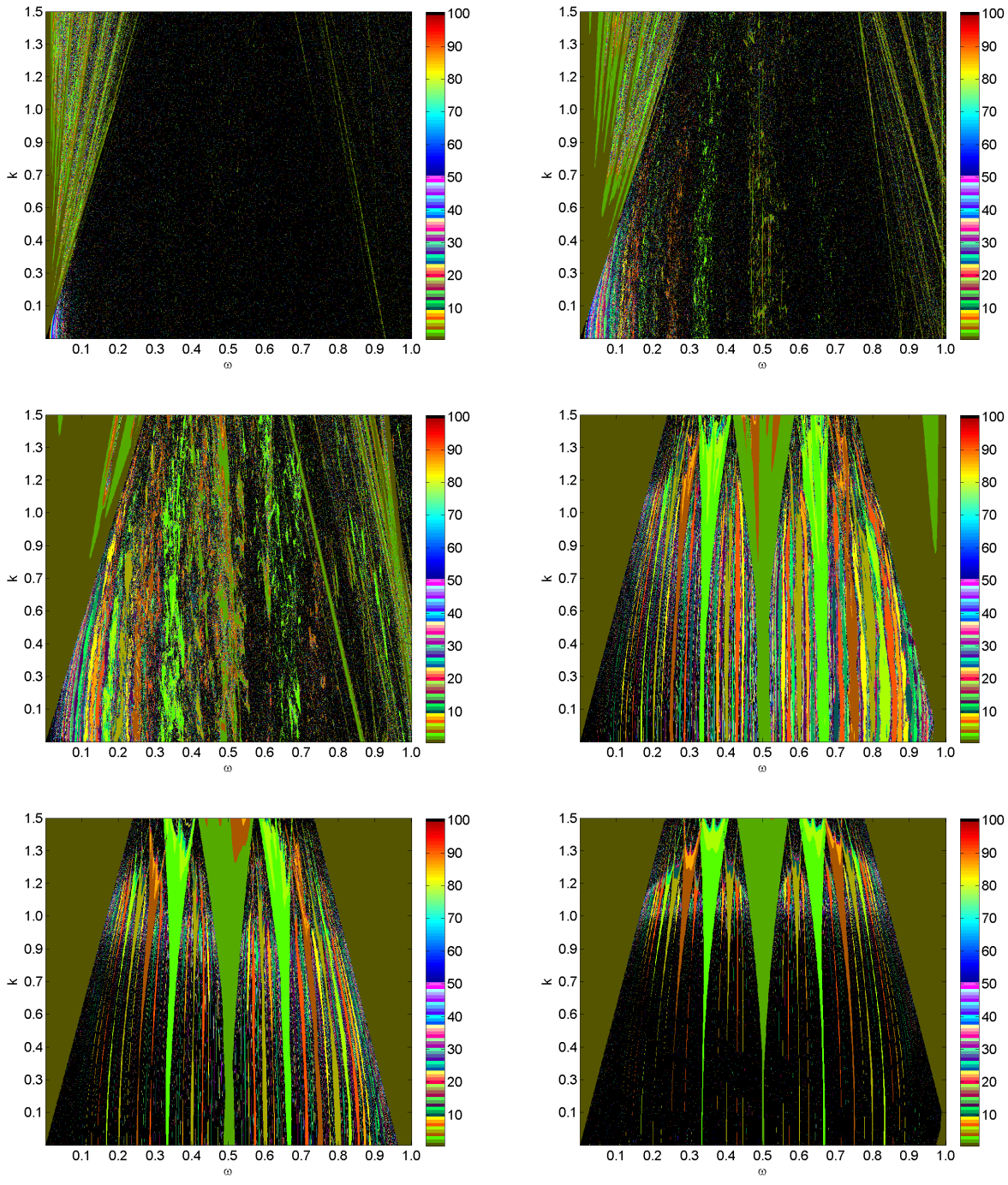


Figure 3.25: The Arnold tongues for $k \in [0, 1.5]$, $\Delta k = 0.0015$, $\omega \in [0, 1]$, $\Delta\omega = 0.001$, $\alpha = 10^{-5}$, $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ and $L_j \in \{0.125, 0.15, 0.175, 0.2, 0.225, 0.25\}$. Δk and $\Delta\omega$ represent the step size in the discretization of k on [0,1.5] and ω on [0,1]. Plots are ordered left to right, and top to bottom. The legend to the right demonstrates the period and corresponding color.

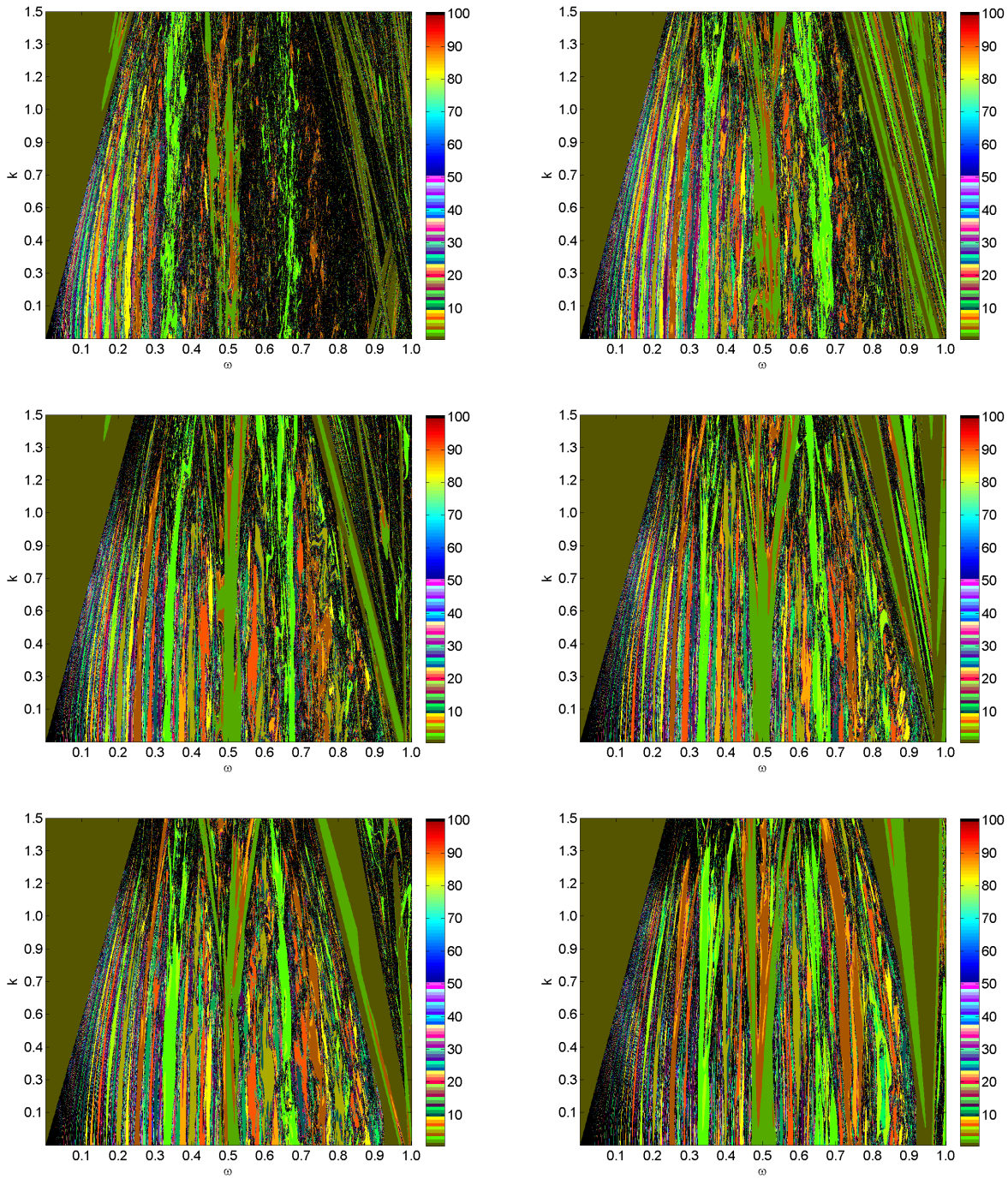


Figure 3.26: The Arnold tongues for $k \in [0, 1.5]$, $\Delta k = 0.0015$, $\omega \in [0, 1]$, $\Delta\omega = 0.001$, $\alpha = \frac{1}{2}10^{-5}$, $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ and $L_j \in \{0.025, 0.05, 0.1, 0.3, 0.5, 0.9\}$. Δk and $\Delta\omega$ represent the step size in the discretization of k on [0,1.5] and ω on [0,1]. Plots are ordered left to right, and top to bottom. The legend to the right demonstrates the period and corresponding color.

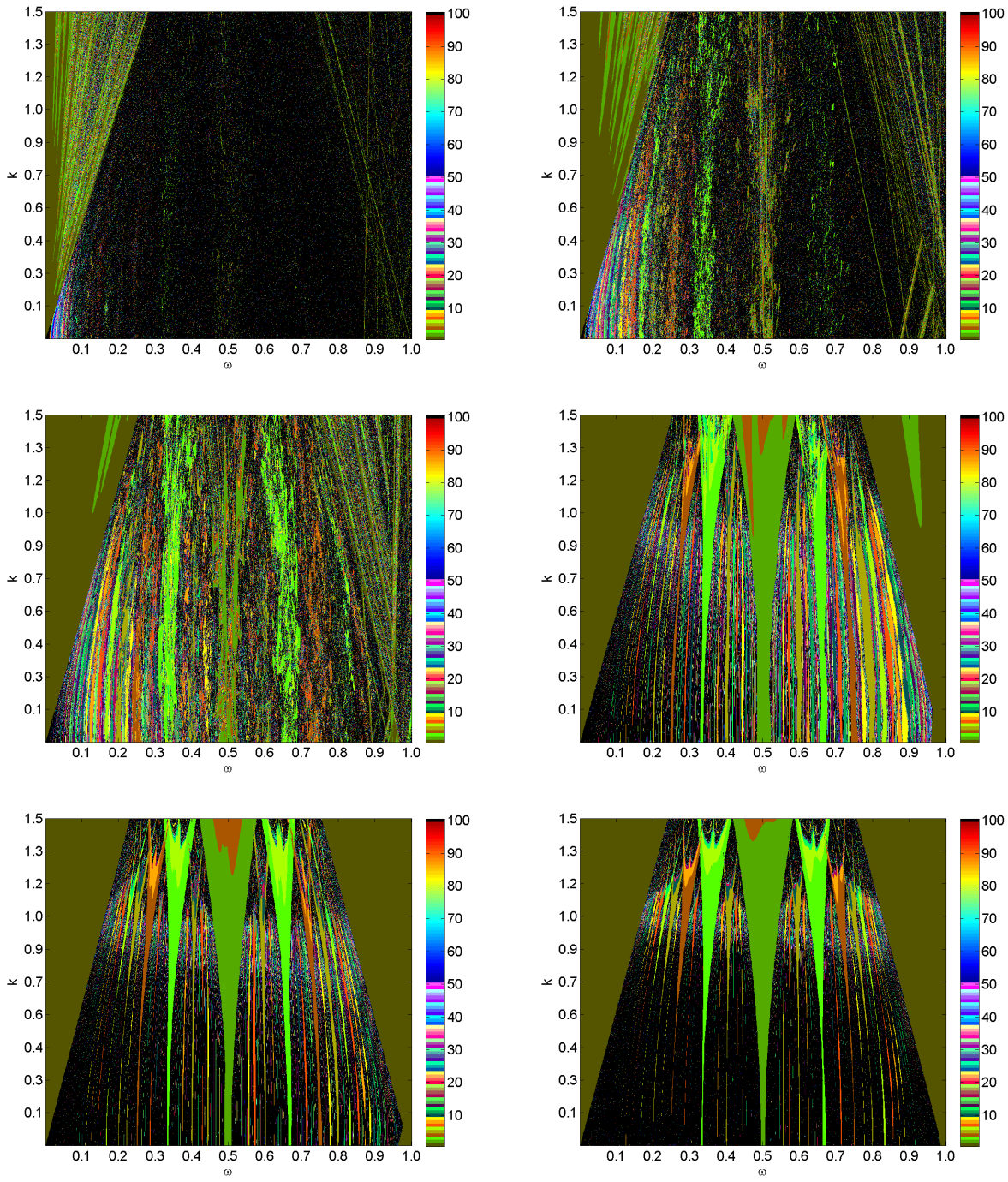


Figure 3.27: The Lyapunov exponent for the deterministic circle map (top left) is compared to the Lyapunov exponent of the random circle map for $L \in \{0.05, 0.1, 0.3, 0.5, 0.9\}$, where $x_0 = 0.7$, $k = 1.5$,$\alpha = 10^{-5}$, and $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ for $\omega \in [0, 1]$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.

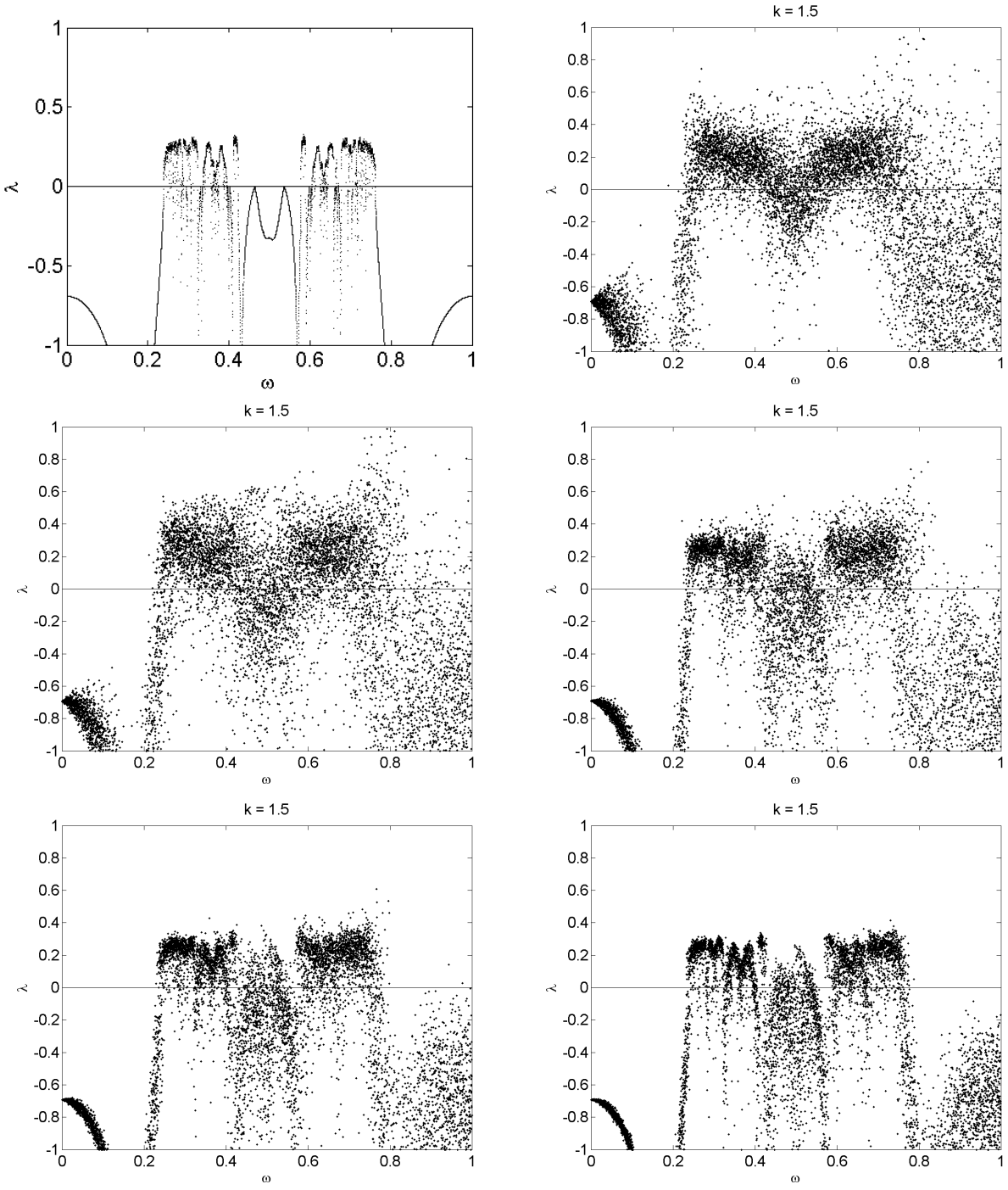
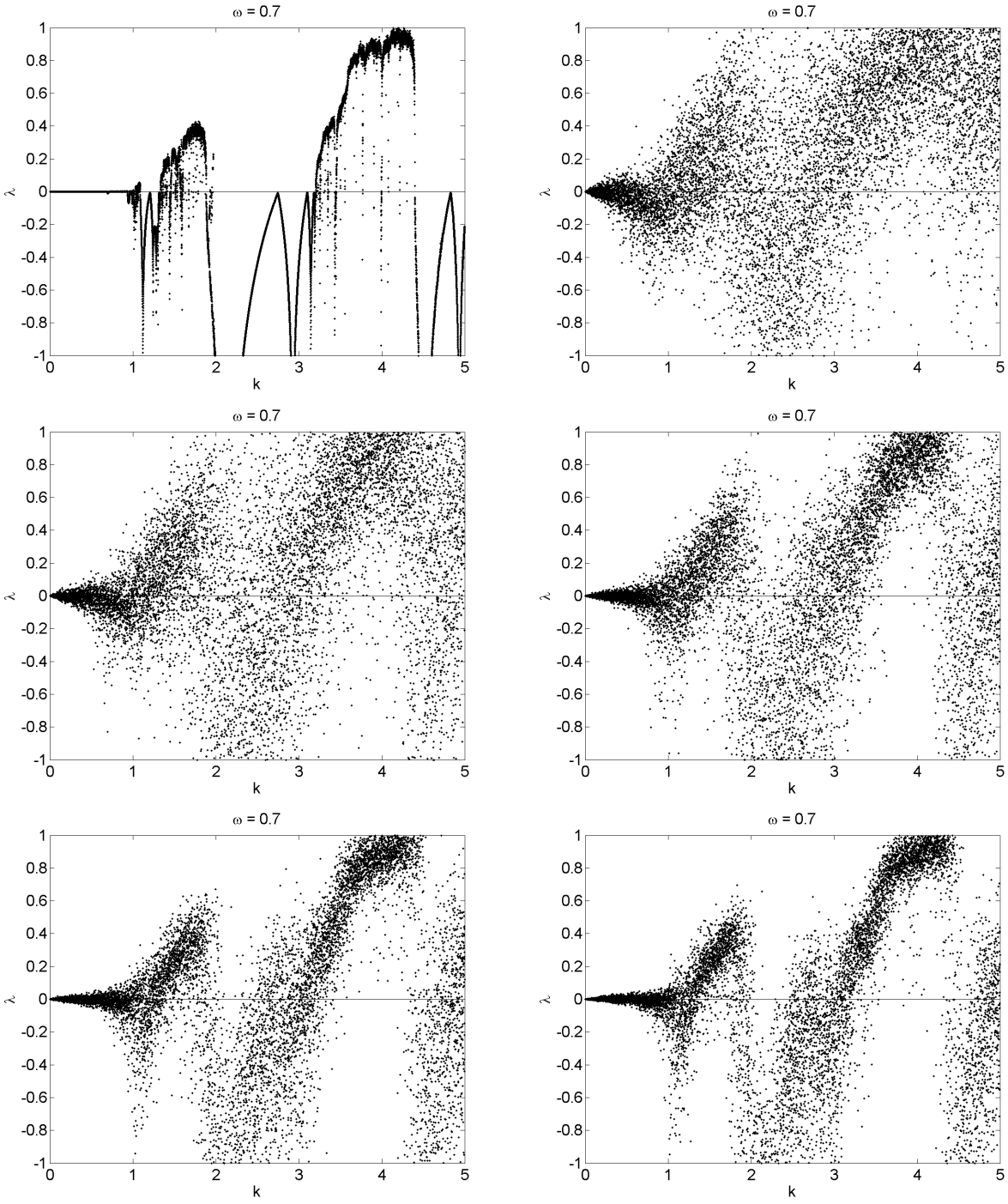
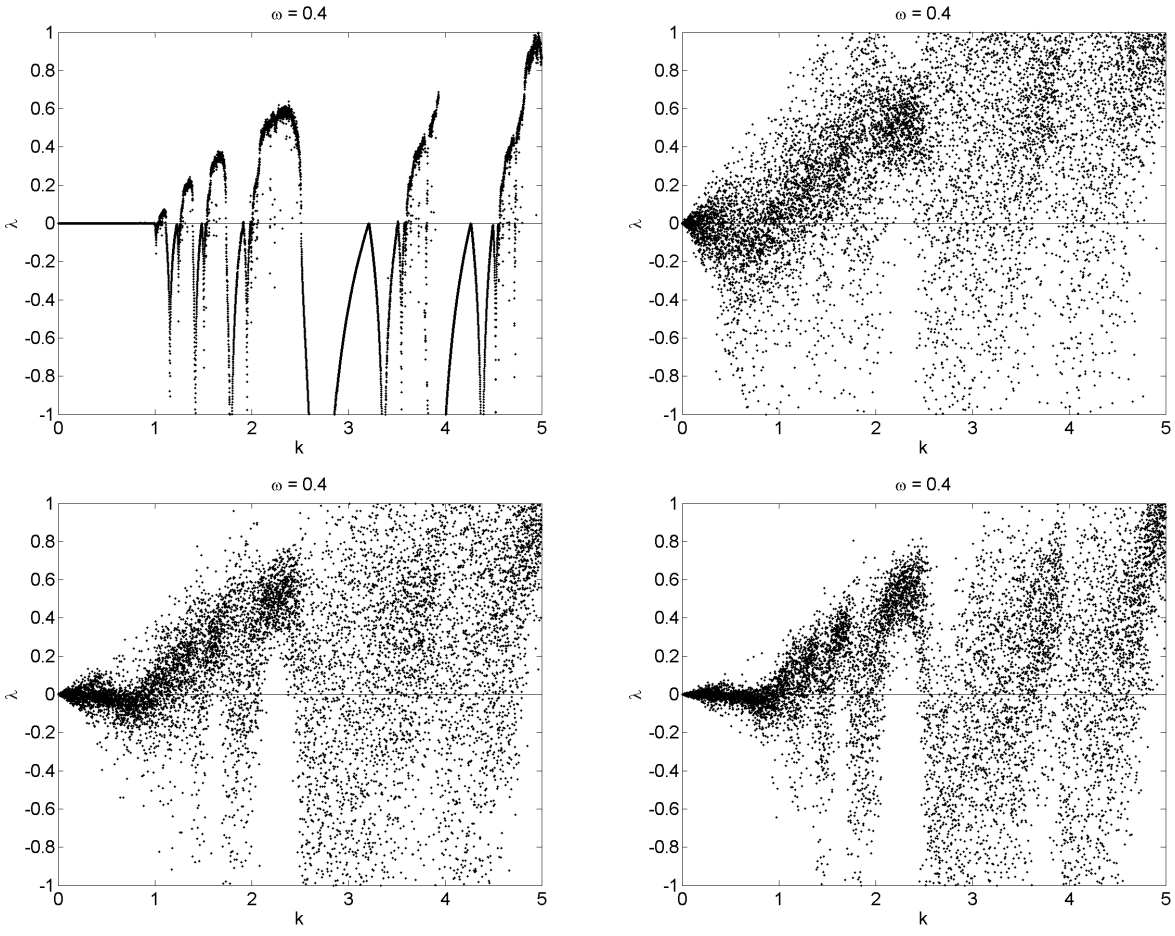


Figure 3.28: The Lyapunov exponent for the deterministic circle map (top left) is compared to the Lyapunov exponent of the random circle map for $L \in \{0.05, 0.1, 0.3, 0.5, 0.7\}$, where $x_0 = 0.7$, $\omega = 0.7$,$\alpha = 10^{-5}$, and $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ for $k \in [0, 5]$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.



Below, the Lyapunov exponent of the circle map for the case where $\alpha = \frac{1}{2}10^{-5}$ demonstrates potentially chaotic behavior in the map (Figure 3.29). There is not much of a qualitative change in the Lyapunov exponent diagrams between the case where the $\alpha = 10^{-5}$ and $\alpha = \frac{1}{2}10^{-5}$ (Figure 3.28, 3.29).

Figure 3.29: The Lyapunov exponent for the deterministic circle map (top left) is compared to the Lyapunov exponent of the random circle map for $L \in \{0.3, 0.5, 0.7\}$, where $x_0 = 0.7$, $\omega = 0.4$,$\alpha = \frac{1}{2}10^{-5}$, and $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ for $k \in [0, 5]$. The number of exponents computed was $N_\lambda = 10,000$. Plots are read left to right, and top to bottom.



Histograms displaying the average fraction of periodic orbits are shown in Figure 3.30 through 3.33. Even though the distribution of period is unknown, we observe that halving α introduces more stable high-period orbits when $L = 0.05, \omega = 0.9, k = 1$ (left side of Figure 3.30 and 3.32). Reducing α

also reduces the frequency of period 6 orbits for $L = 0.9, \omega = 0.6, k = 1.5$ (right side of Figure 3.31 and 3.33).

Figure 3.30: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = 10^{-5}$, $\omega = 0.9$ and $k = 1$. For (L, N), we have $(0.025, 400)$ (left), $(0.05, 200)$ (middle), and $(0.1, 100)$ (right).

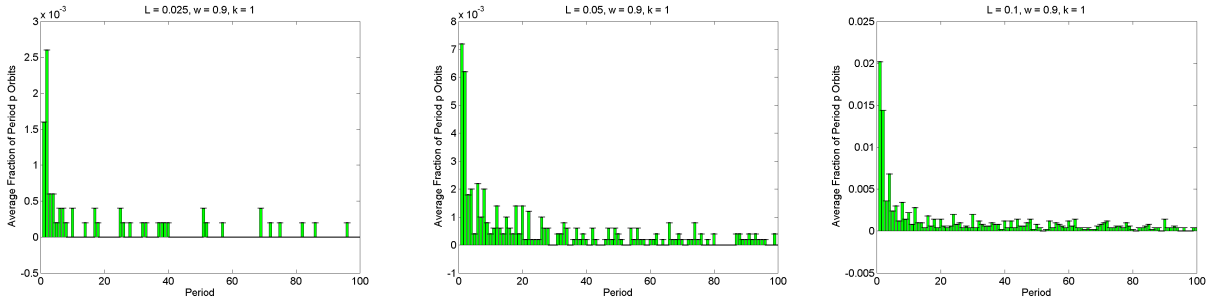


Figure 3.31: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = 10^{-5}$, $\omega = 0.6$ and $k = 1.5$. For (L, N), we have $(0.05, 200)$ (left), $(0.1, 100)$ (middle), and $(0.9, 100)$ (right).

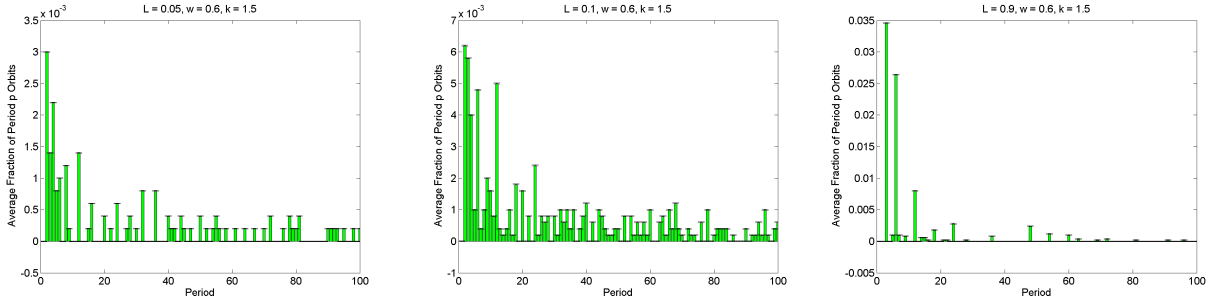


Figure 3.32: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = \frac{1}{2}10^{-5}$, $\omega = 0.9$ and $k = 1$. For (L, N), we have $(0.025, 400)$ (left), $(0.05, 200)$ (middle), and $(0.1, 100)$ (right).

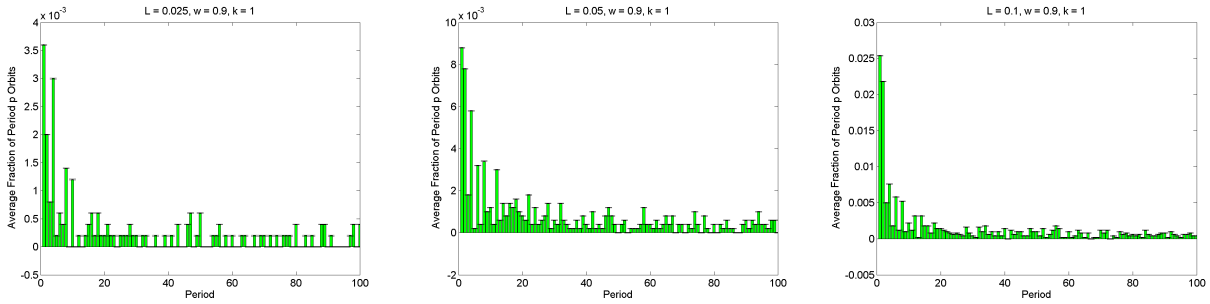


Figure 3.33: Average fraction of period p orbits for the random circle map, where 5000 simulations are plotted. The error bars indicate the standard error of the calculation of the mean. In all plots, $\alpha = \frac{1}{2}10^{-5}$, $\omega = 0.6$ and $k = 1.5$. For (L, N), we have $(0.05, 200)$ (left), $(0.1, 100)$ (middle), and $(0.9, 100)$ (right).

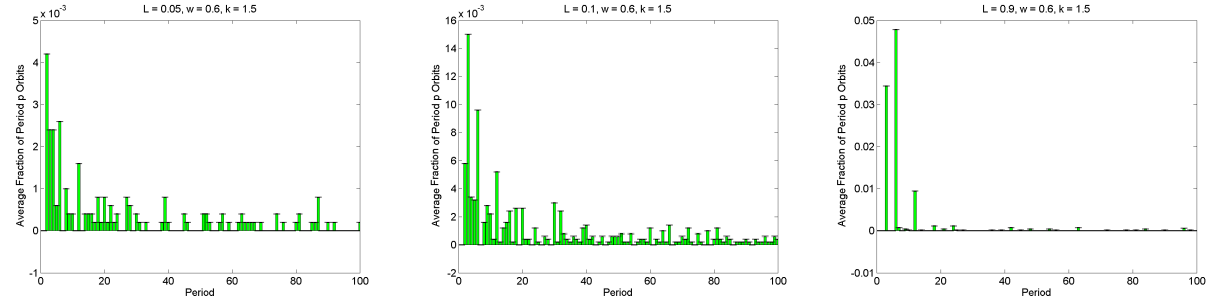


Figure 3.34 and Figure 3.35 are plots of the rotation number of the random circle map for normally distributed Fourier mode amplitudes. The uniform case (Figure 3.21) has a discontinuous section of ρ when $\omega \approx 0.2$ and $L = 0.05$, whereas the normal case shows an almost continuous section of ρ. This may indicate relatively more chaotic behavior in the uniform case. Additionally, the rotation number when $L = 0.1$ (middle of Figure 3.34) appears less noisy than the uniform case (middle of Figure 3.21). In line with this idea, the plots in Figure 3.35 are overall smoother than in Figure 3.35. In all, it appears the normally distributed Fourier mode amplitudes produce

smoother plots of rotation numbers than the uniform case.

Figure 3.34: The devil's staircase for $L = 0.05$ (left), $L = 0.3$ (middle), and $L = 0.5$ (right), where $k = 1$,$\alpha = 10^{-5}$ and $\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$. For small L, the noise is more pronounced than for large L.

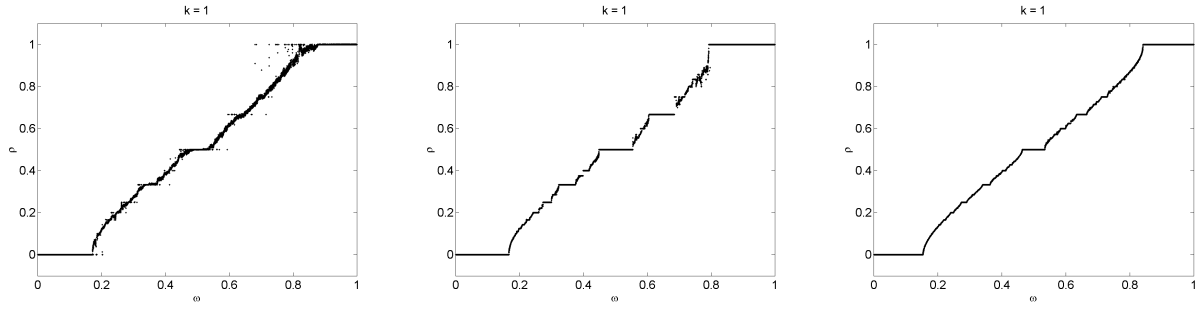
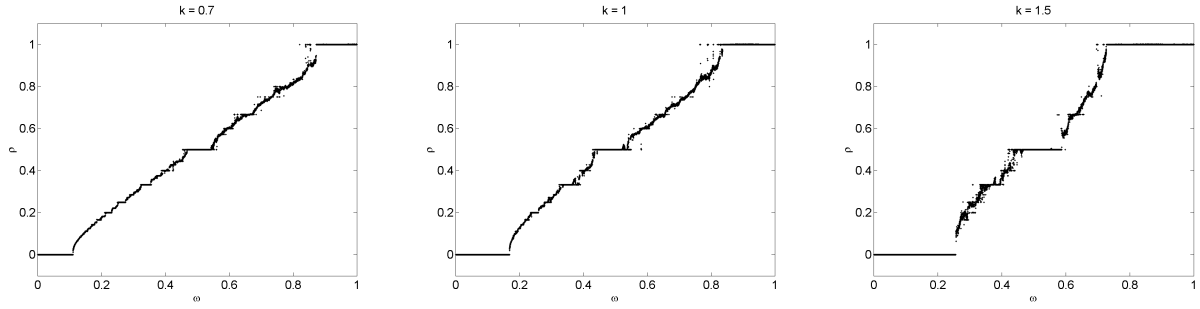


Figure 3.35: The devil's staircase for $k = 0.7$ (left), $k = 1$ (middle), and $k = 1.5$ (right). In each plot, $\alpha = 10^{-5}$,$\hat{\xi}_n \sim N(0, \alpha e^{-L|n|})$ and $L = 0.1$. The discontinuities increase as k grows.



Chapter 4

Conclusion

4.1 Spatially Random Processes and 1-Dimensional Maps

This thesis explored various visualizations of bifurcation diagrams for the logistic and circle map, but establishing a formal way of expressing the bifurcation diagram for spatially random maps would be advantageous for exploring higher-dimensional maps.

Between both the logistic map and the circle map, a commonality is the question: how many period p orbits is the spatially random process responsible for? This question is another way of asking how to find the expected number of zeros of the function $\mathcal{F}(x) = f(x) - x$, where $f(x)$ represents the randomized map. The distribution of zeros of $\mathcal{F}(x)$ would confer a greater understanding of how the spatially random process stabilizes or destabilizes a map. Some prior work in analyzing the number of zeros of a function with a random process based on a Fourier series has been done, but in a slightly different vein; the random process had independent and identically distributed random variables [10]. We are interested in a theoretical explanation of the probability density function of random process for uniformly and normally distributed Fourier mode amplitudes (independent, non-identical). Figure 2.5 suggests $\xi(x)$ is normally distributed when the Fourier modes are drawn from a uniform distribution, but this is far from a generalization. When the modes are normally distributed, we achieve a log-normal spatially random process that mimics the noise in hydraulic modeling. Therefore, the circle map based on normally distributed modes most closely models the fluid flow in the aquifer. Thus, the next step is to delve into the theoretical role a log-normal process plays in the stability of the one-dimensional map, by perhaps exploring

how the log-normal process affects the chaotic basin of attraction in the map, or by finding the expected number of zeros of $\mathcal{F}(x)$.

4.1.1 Logistic Map

The spatially random processes appear to stabilize the logistic map in certain regions of the bifurcation diagram, specifically where $r \in [3, 4]$ (Figure 3.1, 3.2). Evidence of the newly stable regions is the presence of low-period orbits and negative Lyapunov exponents for the random map (Figure 3.4). On the other hand, the presence of high-period orbits for small r in Figure 3.1 and high density of positive Lyapunov exponents (Figure 3.4) suggests the possibility of chaos.

As expected, reducing the noise introduced in the logistic map by halving the variance of the random process resulted in bifurcation diagrams that were more similar to the deterministic case. There were fewer high-period orbits for small values of r and the spread of orbit locations was smaller. However, despite reducing the magnitude of the noise, the random process continues to stabilize the map (Figure 3.3). Negative Lyapunov exponents calculated for this level of variance (Figure 3.5) support the idea of stabilization, however, there was a high density of positive exponents as well, suggesting chaotic tendencies.

The preservation of the negative spike in the plots of Lyapunov exponents for the random map (Figure 3.4, 3.5) may be explained by the construction of the bound on σ in (2.15). As $r \to 4$, the dominant term $\ln(4/r) \to 0$, so the standard deviation of the spatially random process diminishes as r increases. This would cause the random map to adopt more features of the deterministic map for large r. An alternative is to consider constructing another bound on σ that does not fall off as sharply as (2.15).

Although the distribution of period (Figure 3.6, 3.8) appeared exponential, the log-scale plot of the histograms demonstrates an exponential distribution is unlikely, since the shape of the data is nonlinear (Figure 3.7). When halving σ, the general shape of the histogram is retained, although it is scaled a little smaller. It appears the density of stable orbits for $r < 3$ diminishes as L is increased (Figure 3.9, 3.10). Additionally, there are fewer stable high period orbits when σ is small.

4.1.2 Circle Map

In contrast to the bifurcation diagrams of the logistic map, which mostly retained the general shape of the deterministic map, the Arnold tongues (Figure 3.11, 3.24) of the random circle map were asymmetric, and for some values of L, did not resemble tongues at all. This suggests the randomness has an overall destabilizing effect. Figure 3.25 demonstrated that the Arnold tongues for normally distributed $\hat{\xi}_n$ undergo complicated fluctuations when L is increased by $\Delta L = 0.025$ in the range [0.1,0.3].

Examining the Lyapunov exponents of the map for various values of L exposed further asymmetries. Figure 3.13, 3.5 and Figure 3.27 showed a skewed distribution of exponents on the right side of the plots, compared to the left side. The mechanism for this asymmetry remains to be determined.

For uniformly distributed Fourier modes, increasing α seems to increase the number of high-period orbits for certain values of $\omega < 0.2$, eliminate period 2 orbits when $\omega > 0.9$, and reduce the number of period 4 orbits in the center tongue (Figure 3.11, 3.12). The histograms in Figure 3.15 and 3.18 support this idea. For instance, the observed frequency of period 1 and 2 orbits have opposite trends when α is halved. Also, depending on α, one may observe period 3 and 6 orbits dominate the plot evenly, or period 3 is half as frequently observed than period 6. Unlike the random logistic map, it does not seem that the distribution of period is consistent in the circle map when the variance of the Fourier modes is reduced.

Mainly, the Arnold tongue diagrams for the circle map based on the uniform and normal distributions were qualitatively similar, though it appeared the normal case produced smoother plots of rotation numbers (Figure 3.34, 3.35). Perhaps the difference between the normal and uniform cases is best highlighted in the tongue diagrams for $L = 0.3$ (middle left of Figure 3.11 and middle right of Figure 3.24). There are many high-period orbits for $\omega \approx 0.5$ in the uniform case, but very few in the normal case. Also, there is a period 2 tongue on the right side of the plot in the normal case, which is absent in the uniform case. A similarity between the simulations involving

the uniform and normal distributions is that they both produced plots of Lyapunov exponents that seem to have a high density of exponents in the lower right corner (Figure 3.13 and Figure 3.27), but only for the case where one varies ω and fixes k to be constant.

Just as the histograms in Figures 3.6, 3.8 tried to suggest a distribution of periods in the logistic map, kernel density estimation in Figure 3.23 points to the rotation numbers of the map being normally distributed (for a large enough bandwidth) for uniformly distributed $\hat{\xi}_n$. This notion is supported by the results from Figure 3.17 and 3.20, where the average fraction of orbits was shown not to follow an exponential distribution. One difference between the log scale plots is that for $\alpha = \frac{1}{2}10^{-5}$, the range of average fraction of periodic orbits seems wider than for $\alpha = 10^{-5}$.

For the normally distributed $\hat{\xi}_n$, halving α introduces more stable high-period orbits when $L = 0.05, \omega = 0.9, k = 1$ (left side of Figure 3.30 and 3.32). Reducing α also reduces the frequency of period 6 orbits for $L = 0.9, \omega = 0.6, k = 1.5$ (right side of Figure 3.31 and 3.33). Studying the results from the circle map for uniformly distributed $\hat{\xi}_n$ was different from the normally distributed case in terms of quantifying period distribution. First, the observed frequency of period 1 and 2 orbits have opposite trends when α is halved (Figure 3.15 and 3.18). Also, in Figure 3.16, period 3 and 6 orbits dominate the plot evenly, yet in Figure 3.19, period 3 is half as frequently observed than period 6. From this perspective, it does not seem that the circle map with two different random processes is similar at all.

The numerical simulations insinuate distributions of rotation number and period for the circle map, but they leave analytic underpinnings to be desired. A next step would be to use the theory surrounding the distribution of the spatially random process to derive a more general result.

4.2 Future Work

4.2.1 Extension to Higher Dimensions

A natural and intuitive next step would be to introduce a spatially random process to a higher dimensional set of differential equations, such as the two-dimensional Lotka-Volterra system

for competitive species interaction

$$\begin{aligned} \dot{x} &= x(a - bx - cy), \\ \dot{y} &= y(d - ex - fy). \end{aligned} \tag{4.1}$$

In this system, the parameters a, b, c, d, e, f are positive constants. A future work could replace one of the parameters with a spatially random process. Limit cycles arise in analyzing this system of differential equations, and it would be interesting to explore the effect of spatial perturbations on these cycles. The system of differential equations is more relevant to the fluid flow problem, but another possible choice is the two-dimensional discrete time dynamical system, the Hénon map,

$$\begin{aligned} x_{n+1} &= 1 - ax_n^2 + y_n \\ y_{n+1} &= bx_n. \end{aligned} \tag{4.2}$$

One might replace one of the constants a or b with a spatially random process. This map was meant to be a simplified model of the Poincaré section of the Lorenz model.

4.2.2 Basin of Attraction for Chaotic Trajectories

Exploring the basin of attraction for chaotic regions of the two maps may offer some interesting insights on how best to initialize remediation. Specifically, we would like to explore the question: which set of initial conditions lead trajectories to chaotic behavior instead of stable orbits? The basin of attraction may offer implications on how one should inject treatment solution in the aquifer to get the best (chaotic) mixing. If a formal description of the onset of chaos for these types of systems were to exist, it would have implications for studying higher-dimensional systems.

One way to quantify whether a trajectory is chaotic is to calculate the Lyapunov exponent, so a future study may attempt to calculate the probability of a positive Lyapunov exponent for any given initial condition. The probability of a positive Lyapunov exponent could be calculated by averaging the observed exponents for many simulations of the random map. This calculation would be a quantitative measure that may distinguish between the images of Lyapunov exponents from Chapter 3, as they are currently only qualitatively assessed.

4.2.3 Stabilizing Effects of Spatial Randomness

An unexpected result of the numerical simulations of the logistic map was the presence of stable low-period orbits where $r \in [3, 4]$. Previously, this region was unstable for the deterministic map. A natural question that arises is: how small can we make the variance of the spatially random process such that this stable region is preserved? This thesis explored using the maximum upper bound $\sigma = \sigma_{max}$ and half of the maximum $\sigma = \frac{1}{2}\sigma_{max}$ from the expression in (2.15). Looking at smaller values of σ, or choosing an alternative method of bounding the variance would be a thought-provoking future study. The circle map also displayed stable period 2 orbits where there were previously only period 1 orbits for $L = 0.3$, but this region disappeared for $L \leq 0.1$ and $L \geq 0.5$. It would be interesting to find the largest $\epsilon > 0$ such that $L = 0.3 \pm \epsilon$ results in the birth of this period 2 region in the circle map. Further study of the stabilized regions in both maps may yield a theoretical explanation for this behavior. The idea of using random processes to stabilize chaotic behavior is a largely unexplored research topic of high interest and potential applications. For example, Hitczenko and Medvedev derived a condition to stabilize weakly unstable equilibria in temporally random processes [8].

4.2.4 Dynamic Load Balancing on the Supercomputer

The original implementation of the numerical simulations was written in MATLAB, and quite inefficient in certain places. We explored and simulated the spatially random logistic map using the dynamic load balancing tool on Janus[1] , the University of Colorado's supercomputer [17]. The recursive nature of the map prevents the individual calculations of an orbit from being parallelized, but a set of iterations may be load balanced over many cores. Improvements to this project include: adapting the program to produce other types of plots, extending the program to model the Arnold Circle Map, and optimizing the post-simulation data processing. Furthermore, we can explore using Newton's method to find the fixed points of the randomized maps. The advantages of this method

[1] Janus has 1368 compute nodes, and each node has 12 processors. Each processor is capable of independently carrying out a series of computations.

over simply iterating the maps would be quicker convergence to the roots and identification of unstable orbits. The current iterative method only locates stable orbits, and its rate of convergence is highly variable.

Appendix A offers more information regarding background information on load balancing tools, program design, and effects of the load balancing tool on performance.

Bibliography

[1] K.B. Athreya and Jack Dai. Random logistic maps. Journal of Theoretical Probability, 13(2):595–608, 2000.

[2] Kendall E. Atkinson. An Introduction to Numerical Analysis. John Wiley & Sons, Inc., 1989.

[3] Patrick Billingsley. Probability and Measure. John Wiley and Sons, Inc., 1995.

[4] Robert L. Devaney. An Introduction to Chaotic Dynamical Systems. The Benjamin/Cummings Publishing Co., Inc., 1986.

[5] The Groundwater Foundation. The basics, 2014.

[6] Lynn W. Gelhar. Stochastic Subsurface Hydrology. Prentice-Hall, Inc., 1993.

[7] Geotechdata.info. Soil permeability coefficient, 2015.

[8] Pawel Hitczenko and Georgi S. Medvedev. Stability of equilibria of randomly perturbed maps. eprint arXiv:1503.05979, 2015.

[9] International Water Management Institute. Advection - dispersion, 2014.

[10] J. P. Kahane. Some Random Series of Functions. Cambridge University Press, 1985.

[11] Jeroen S.W. Lamb, Martin Rasmussen, and Christian S. Rodrigues. Topological bifurcations of minimal invariant sets for set-valued dynamical systems. Proceedings of the American Mathematical Society, 2013.

[12] Robert May. Simple mathematical models with very complicated dynamics. Nature, 261:459–67, 1976.

[13] James D. Meiss. Differential Dynamical Systems. Society for Industrial and Applied Mathematics, 2007.

[14] Roseanna M. Neupauer, James D. Meiss, and David C. Mays. Chaotic advection and reaction during engineered injection and extraction in heterogeneous porous media. Water Resources Research, 50:1433–47, 2014.

[15] North Carolina Division of Water Resources. Basic hydrogeology, 2014.

[16] S. Neil Rasband. Chaotic Dynamics of Nonlinear Systems. John Wiley & Sons, Inc., 1990.

[17] Research Computing at University of Colorado Boulder. Load balance tool, 2014. https://www.rc.colorado.edu/node/507.

[18] Sheldon Ross. A First Course in Probability. Pearson Education, Inc., 2010.

[19] United States Geological Service. Groundwater use in the united states, 2015.

[20] Steven H. Strogatz. Nonlinear Dynamics and Chaos. Perseus Books Publishing, LLC, 1994.

[21] The HDF Group. Hdf5 user's guide, 2014. http://www.hdfgroup.org/HDF5/doc/UG/index.html.

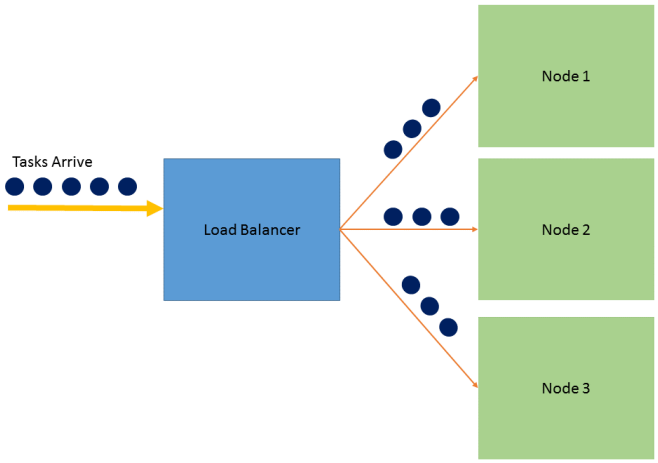
[22] Marc H. Willibeek-LeMair and Anthony P. Reeves. Strategies for dynamic load balancing on highly parallel computers. IEEE Transactions on Parallel and Distributed Computing, 4(4):979–93, September 1993.

Appendix A

Load Balancing Tool and Detailed Program Description

A **load balancer** is a tool that dynamically reassigns tasks as the processors complete their work. There are many strategies for load balancing, such as sender initiated diffusion, receiver initiated diffusion, hierarchical balance model, etc. [22]. The Load Balance tool on Janus uses a master-slave strategy for balancing [17]. In general, a load balancer recognizes the number of processors that are going to be used in a simulation, and manages the workload distribution among them. Figure A.1 demonstrates the load balancer coordinating tasks over three nodes. If, for

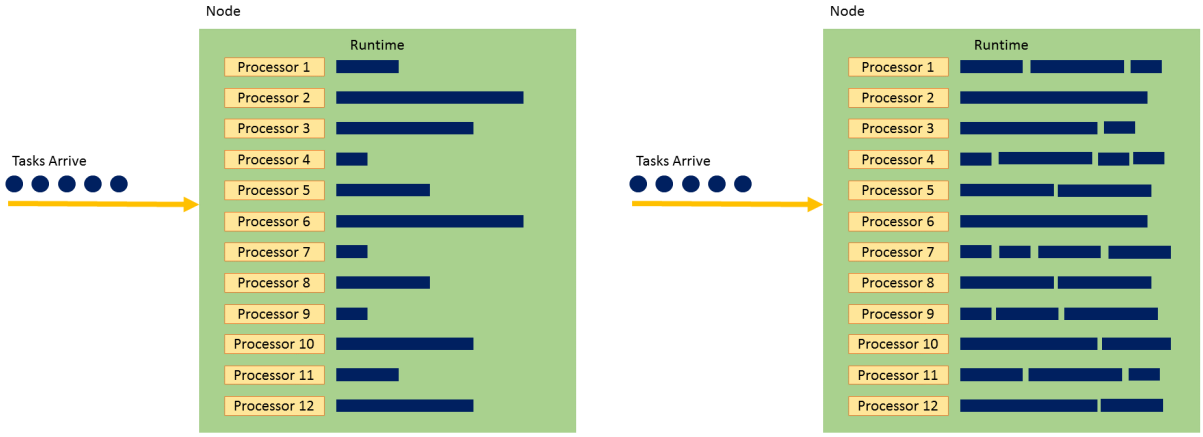
Figure A.1: Load balancing example over three nodes. The load balancer is invoked over 3 nodes; it determines the work distribution over each node.



example, Processor A finishes its load early (perhaps its initial condition led to near-immediate

convergence), the load balancer assigns Processor A more work by reducing the queue of tasks on Processor B (a processor taking more time to complete its current task) and passing some tasks to Processor A. Figure A.2 shows how the load balancer reassigns tasks between the 12 processors on any given node. Figure A.3 demonstrates an example of a simulation using 40 nodes on Janus,

Figure A.2: As tasks arrive, each processor in a node takes an initial task (left). As the processors finish their task, the ones who finish first are assigned more work while the others continue running (right).



where there are 12 processors per node. Below is an outline of how each processor is called in the program.

(1) Each processor should take a different initial x_0 and report whether the initial condition led to finding a stable orbit

- The load balancer determines work distribution over the processors.

(2) Repeat the above step for a large number of different random maps $R_i(x)$, $i = 0, 1, 2, ...\bar{N}$ in order to find the sample mean of any given period p orbit and the orbit locations (if there was convergence).

- Use the data to produce a histogram of periodic orbits that depicts the expected number of period p orbits for the random map
- Create bifurcation diagrams for various values of L.

- Use a HDF5 file to store the simulation data

The original simulation was written in serial code in MATLAB, and had extremely variable run times (due to the nature of fixed point iteration). The original version of the code was rewritten for efficiency, speed, and scalability. The new implementation, written in C++ and Python, was capable of reproducing two types plots: histograms of observed periodic orbits and bifurcation diagrams for varying values of the correlation length, L. Figure A.4 is a work flow of the program structure, and we offer a detailed program description below.

The program begins with `generate_cmdlines`, where the user specifies the desired parameters for the simulation. The resulting data may be processed and visualized as a histogram of periodic orbits or bifurcation diagram.

(1) `generate_cmdlines`: the set up file for the simulation; specifies the parameters used in the map realizations. It will generate all the bash script files needed by Janus's work scheduler, `slurm`.

(2) bash scripts: Each of these scripts will invoke `generate_rands`, based on parameters given in `generate_cmdlines`. `generate_rands` creates a data file of values of a_n, b_n, the Fourier modes of $\xi(x)$. The script `myfunc` uses the output from `generate_rands`.

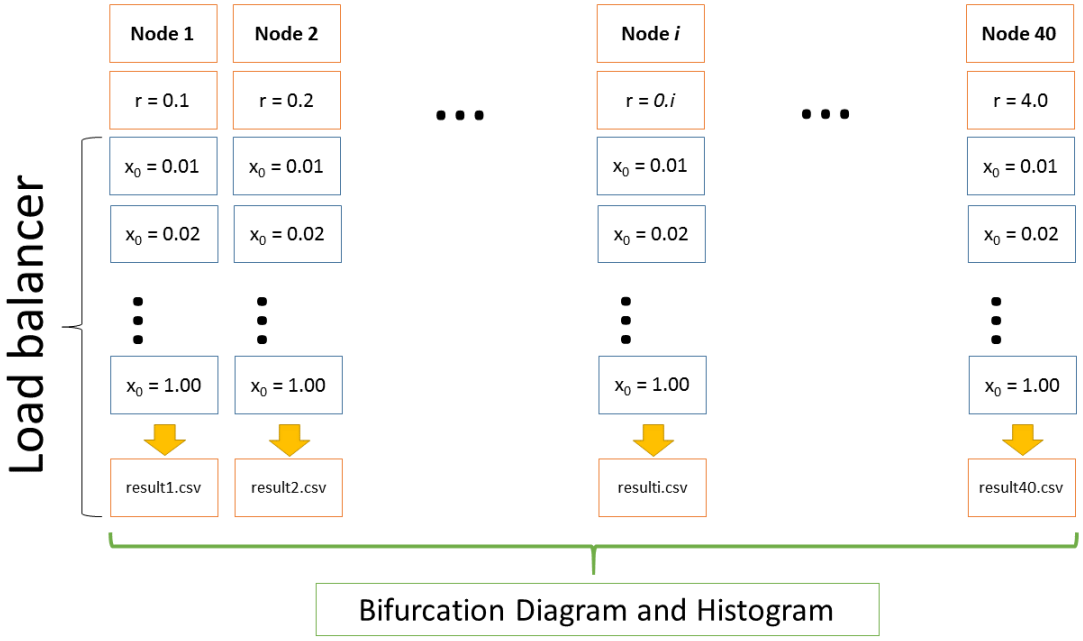
(3) `generate_rands`: Generate randomized parameters a_n, b_n and write them to file for the parameters specified by `generate_cmdlines`.

(4) `myfunc`: Iteratively solve the map $f(x) = x$, and print out the orbit locations if they exist or return nothing if the map diverges. All output from `myfunc` will be fed into a `result` file associated with the given bash script.

(5) `csv2hdf5`: Convert `result` to a HDF5 file while checking for uniqueness in the data set. Save the processed data in an HDF5 file for archival purposes.

(6) `unique`: Check for uniqueness in the data set and create a histogram of the data

Figure A.3: Load Balancing Tool Overview: The load balancer is invoked over 40 nodes, where each node handles some value of $r \in [0, 4]$ and a number N_x of initial conditions $x_0 \in [0, 1]$ are tested. Each node produces a datafile, which is compiled with the other results to create a bifurcation diagram or histogram of periodic orbits.



(7) `plotbif`: Use the HDF5 file to produce the bifurcation diagram.

The HDF5 file format was used to store the simulation results in a better archival format. Single core optimization techniques, such as SIMD loop vectorization and function in-lining, as well as using a dynamic load balancer for more efficient work distribution were applied. The following are the optimization techniques applied:

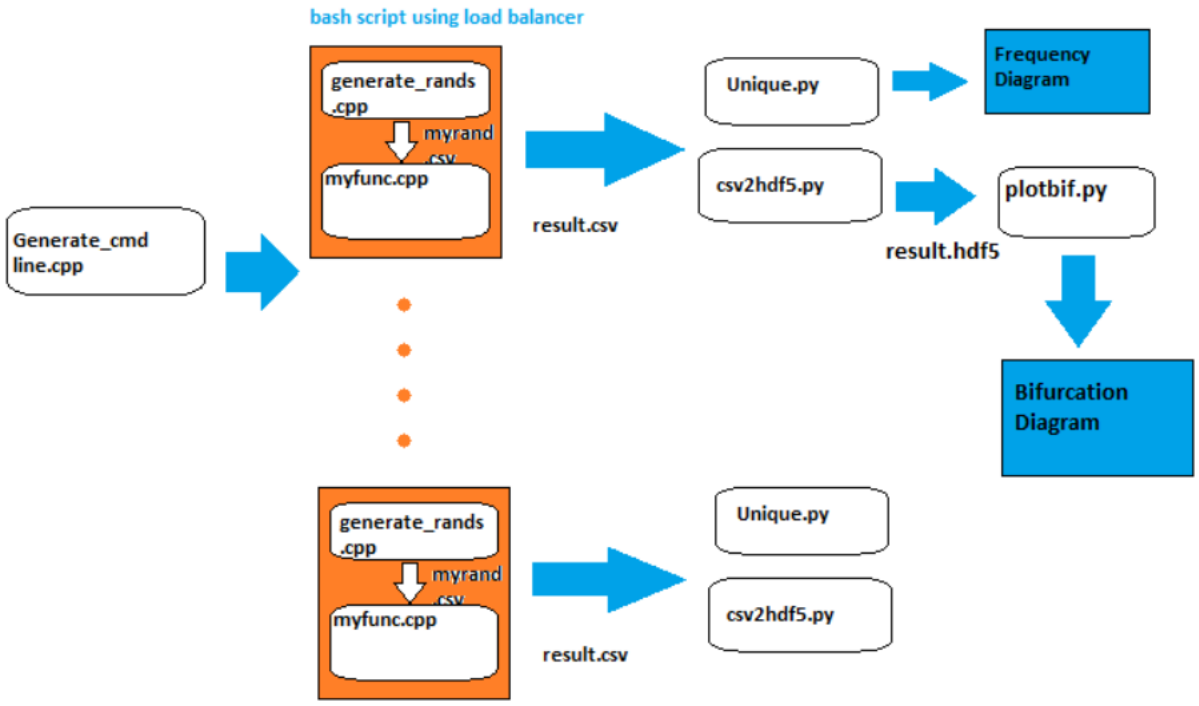
- Preferential use of the multiply and add operators where possible, since they are less expensive than divide and subtract operators
- Used a reduction on the loop that computes the Fourier Series in order to take advantage of the data parallelism with SIMD
- Loop structure was reorganized to take advantage of C++ being row-oriented (outer loop should go by rows, then inner loops go by columns)
- Functional in-lining in the C++ code to reduce the number of function calls
- The lack of a built in uniform random number generator that generates a random double between two doubles led to the creation of a pseudo random number generator with the use of `rand` and `srand`.

The benchmarking (strong scaling study) results (Figure A.5) imply the best speedup and efficiency is gained when invoking the load balancer on one node (12 processors), although we tested our simulation over 16 nodes (192 processors). The formulas for calculating speedup S and efficiency E are

$$\begin{aligned} S &= \frac{t_s}{t_l} \\ E &= \frac{S}{n_p}, \end{aligned} \tag{A.1}$$

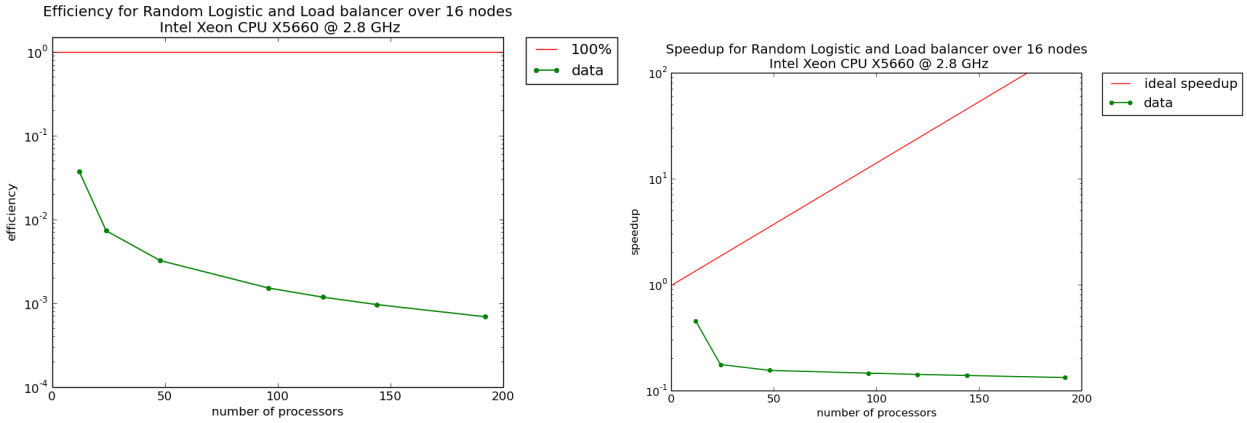
where t_s is the serial computation time, t_l is the load balanced computation time, and n_p is the number of processors used. However, since the original serial implementation in MATLAB could take days to complete, the speedup and efficiency in Figure A.5 was computed using a serial

Figure A.4: Load balancing work flow: The load balancing tool on Janus takes as input a list of command line prompts (created by `generate_cmdline`) calling the executable files `myfunc` and `generate_rands`. `generate_cmdline` specifies the parameters the user intends to test for the simulation. Each node produces a file called `result`, which is parsed by `Unique` and `csv2hdf5` to get a set of unique orbits to store in a HDF5 file. `Unique` also creates the histogram. The final script, `plotbif`, generates the bifurcation diagram.



implementation in C++, so these plots reflect speedup and efficiency where single core optimization has already been applied. The diminishing speedup and efficiency for larger numbers of processors is likely due to the overhead cost of coordinating tasks over many processors and nodes.

Figure A.5: Efficiency (left) and Speedup (right) of the new implementation. The best efficiency occurred for one node, and the best speedup was also achieved for 1 node.



Appendix B

MATLAB Code

B.1 Logistic Map

B.1.1 Iterating the Logistic Map

```
% Make a cobweb plot for a 1D difference equation map
% R0 parameter, 0 <= R <= 4
% x0 Initial condition
% N Number of iterations

% Amy Le
% May 29, 2014
% vectorized call to R(x) on 3/13/15

function [xv, t, mymap] = cobweb(x0, iter, a, b, r, N, xlen)
if length(x0) == 1   % iterate the logistic map
    x = zeros(iter+1,1);
    k = 1:N;
    x(1) = x0;

    for ic = 1:iter
        xtmp = x(ic)*ones(1,N);
        res = 2*(a'.*cos(2*pi*k.*xtmp) - b'.*sin(2*pi*k.*xtmp));
        myR = exp( log(r) + sum(res) );
        x(ic + 1) = myR * x(ic) * (1 - x(ic));
    end
    if xlen ~= 0
        xv = x(end-xlen+1:end);
    else
        xv = x;
    end
    t = 0;
    mymap = 0;

else % the function at every place in [0,1]
    t = x0;
    myR2 = zeros(length(t),1);
    for j = 1:length(t)
```

```
        myR2(j) = R(t(j),a,b,r,N);
    end
%     disp([t', x0', myR2, a, b])
    mymap = myR2'.*(t.*(1-t));
    xv = 0;

    % % plot cobweb
    plot(t,mymap,'Linewidth',2)
    hold on
    axis('square'); axis([0 1 0 1]);
    set(gca,'XTick',(0:0.2:1),'YTick',(0:0.2:1))

    % % plot the line y = x
    fplot('1*y',[0 1],'r');

    x = zeros(iter,1);
    x(1) = 0.5;
    for ic = 1:iter
        myR(ic) = R(x(ic),a,b,r,N);
        x(ic + 1) = myR(ic) * x(ic) * (1 - x(ic));
    end

%     line([x(1) x(1)],[0 x(2)],'Color','g')
    for ic = iter-100:iter-1
        line([x(ic) x(ic+1)],[x(ic+1) x(ic+1)],'Color','g')
        line([x(ic+1) x(ic+1)],[x(ic+1) x(ic+2)],'Color','g')
        plot(x(ic+1), x(ic+1),'k*');
    end
    line([x(iter) x(iter+1)],[x(iter+1) x(iter+1)],'Color','g')
    xlabel('x_n')
    ylabel('x_{n+1}')
%     title('Random Cobweb Diagram')
    set(gca,'FontSize',15)
    set(findall(gcf,'type','text'),'FontSize',15)
    set(findall(gca, 'Type', 'Line'),'LineWidth',2);
end
end



% removed function call to S.m on 2/19/15
% vectorized on 3/13/15

function [av, bv, sigma, alpha] = myrand(L,N,r)
a = zeros(N,1);
b = zeros(N,1);
sigma = log(4/r) * (sqrt(2/3)) * ( tanh(L*0.25) / sqrt(tanh(L*0.5)) );
sigma = 0.5*sigma;
alpha = sigma^2 * tanh(L/2);
myindex = 1:N;
Mn = sqrt( 1.5 * ( alpha * exp(-L * abs(myindex)) ) );
for j = 1:N
    %     pd = makedist('Normal','mu',mu,'sigma',stddev);
    pd = makedist('Uniform','lower',-Mn(j),'upper',Mn(j));
    a(j) = random(pd);
    b(j) = random(pd);
end
```

```
av = a;
bv = b;
end



% get unique orbits
% input
% orbits = array of all orbit orders + locations
% tol = tolerance

% output
% w = array of orbit orders + unique orbit locations

function [ w ] = uorbits( orbits, tol )
orbits = sort(orbits,2,'descend'); % sort orbits by row in desc order
u_orbits = unique(orbits,'rows');  % delete repeating rows
[row, col] = size(u_orbits);

% delete more rows based on tol
for i = 2:row
    if u_orbits(i,1) == u_orbits(i-1,1)
        if norm(u_orbits(i,:) - u_orbits(i-1,:)) <= tol
            u_orbits(i-1,:) = ones(1,col)*-1;
        end
    end
end

% delete the -1 rows
w = unique(u_orbits,'rows');
if w(1,1) == -1   % if the first element is -1, delete it
    w = w(2:end,:);
end
end



% histogram of Tm/sm
clc; clear; close all

L = 0.1;
N = 100;
r = 3.3;
nvec = 1:N;
maxiter = 1000;
myvar = zeros(maxiter,1);

for k = 1:maxiter
[av, bv, sigma, alpha] = myrand(L,N,r);
Tm = 2*sum(av);
sm = sqrt(0.5*alpha + alpha*sum(exp(-L*nvec)));
myvar(k) = Tm/sm;
end

%pdf of normal RV
x = linspace(-6,6,1000);
```

```
y = normpdf(x,0,1);

hist(myvar)
axis([-6 6 0 300])
hold on
plot(x,650*y,'r','Linewidth',3.5)
h = findobj(gca,'Type','patch');
set(h,'FaceColor',[0.5 0 0.5],'EdgeColor','k')
xlabel('Number of Observations')
ylabel('Frequency')
set(gca,'FontSize',16)
set(findall(gcf,'type','text'),'FontSize',16)
```

B.1.2 Upper and Lower Bounds on the Map

```
% bound the map at each position

function envelope(r,N,L,xlen)
iter = 1000;    % number of x locations
kmax = 500;    % number of different realizations
st = linspace(0,1,kmax);    % vector of initial conditions

myextrema = zeros(kmax,3);
myextrema(:,1) = st';
myextrema(:,2) = ones(kmax,1);
allmydata = zeros(kmax,kmax);

for k = 1:kmax
    [a,b] = myrand(L,N,r);    % make a new random draw for ea realization
    [~, ~,mymap] = cobweb(st, iter, a, b,r,N,xlen);
    allmydata(k,:) = mymap;
    if k == 250
        plot(st,mymap,'r','Linewidth',2)
        hold on
    elseif k == 475
        plot(st,mymap,'r','Linewidth',2)
    end

end

myextrema(:,2) = min(allmydata)';
myextrema(:,3) = max(allmydata)';
% csvwrite(strcat(myname, '.csv'),myextrema)
plot(myextrema(:,1),myextrema(:,2),'k',myextrema(:,1),myextrema(:,3),'k','↩
    Linewidth',2)
xlabel('x_n')
ylabel('x_{n+1}')
axis([0 1 0 1])
% title('Upper and lower bounds on the random logistic map')
set(gca,'FontSize',15)
set(findall(gcf,'type','text'),'FontSize',15)
h=gcf;
name = ['envelope_',num2str(kmax),'_r',num2str(r),'_L',num2str(L),'.png'];
path = 'C:\Users\amy\Dropbox\thesis\logistic_map_code\figures\';
```

```
saveas (h, [path,name], 'png');
end
```

B.1.3 Bifurcation Diagram

```
% bif_driver
% main function to call bifurcation function for logistic map
% 3/13/15

clear; clc; close all

% set up params
nr1 = 800;%0;
nr2 = 1000;%0;
nx = 1000;
nL = 10;
maxp = 256;
pth = 'C:\Users\swamy\Documents\amy\thesis\logistic map\bifurcation_data\';

% other params
mdname = 'rlog_bif_metadata_3_15_15.csv';
froot = 'rlog_bif_3_15_15_L';
tol = 10e-6;
N = 100;                        % number of Fourier modes
dr1 = (3-tol)/nr1;
dr2 = ((4-tol) - (3+tol))/nr2;
L = linspace(0+tol,1-tol,nL);
r = [linspace(tol,3,nr1),linspace(2.5+tol,4-tol,nr2)];
rlen = length(r);
x0 = linspace(tol, 1-tol, nx);          % initial conditions
sigma_vec = zeros(length(r),1);
numcols = maxp+2; % [r, orbit_order, orbit_locations]

%[L, dr1, dr2, N, minsigma, maxsigma, nx]
metadata = zeros(nL,7);

for i = 5:length(L)
    disp(i);
    res_data = zeros(rlen*nx,numcols); %container for data
    ressum = 1; %keep track of index in container

    for j = 2:rlen
        [a,b, sigma, alpha] = myrand(L(i),N,r(j));
        sigma_vec(j) = sigma;
        res = bifur(r(j), N, a, b, x0, maxp, tol);
        [rres,~] = size(res);
        if rres >= 1
            res_data(ressum:ressum+rres-1,:) = res;
        end
        ressum = ressum + rres;
    end

    % delete the empty space, if it exists
    tmp = res_data(:,1) == 0; %look for the first zero element
```

```
    res_ind = find(tmp,1) - 1; %grab index
    if isempty(res_ind) == 0
        res_data = res_data(1:res_ind,:); %remove unfilled entries
    end

    % make one large write operation
    tmpname = [froot,num2str(L(i)),'.csv'];
    dlmwrite([pth,tmpname], res_data, '-append');

    % plot diagram
    plotbif(pth,tmpname, L(i), length(r), length(x0), maxp)

    % record metadata for later reference
    metadata(i,:) = [L(i), dr1, dr2, N, min(sigma), max(sigma), nx];
end

% write metadata for all sims
dlmwrite([pth,mdname], metadata);



% compute the probability of a P1, P2, etc. orbit
% input
% r = param in [0,4]
% N = num fourier modes, usually 100
% a,b = rand vecs based on prob dist
% x0 = intial condition
% maxp = max num period to look for
% tol = tolerance

% output
% myorbs = array:
% myorbs(:,1) = r,
% myorbs(:,2) = period order
% myorbs(:,3:maxp) = unique orbit locations for (r,L,x0) tuple

function myorbs = bifur(r, N, a, b, x0, maxp, tol)
trunc = 2*maxp;   % truncate the list of iterates at this length
iter = 1000 + maxp; % number iterations of cobweb for one x0
orbits = -1*ones(length(x0),maxp+1);
u = 1;


for i = 1:length(x0)
    [xv, ~, ~] = cobweb(x0(i), iter, a, b, r, N, trunc);
    for w = 1:maxp    % check period of orbit
        if abs(xv(end-w) - xv(end)) < tol
            orbits(u,1) = w;    % w is period order
            orbits(u,2) = xv(end);        % xv has orbit location

            for v = 3:w+1    % get higher order period orbit locations
                orbits(u,v) = xv(end-v+2);
            end

            u = u + 1;    % row counter
            break
        end
```

```
    end
end


myorbs = uorbits(orbits, tol);
[urow, ucol] = size(myorbs);
if isempty(myorbs)
    myorbs = [];%[r, -1*ones(1,ucol)];  % record divergence for this r
else
    myorbs = [r*ones(urow,1), myorbs];
end

end




% plot the bifurcation diagram for the rand log map
% read the data file in fname line by line to plot

function plotbif(pth, fname, L, rlen, x0len, maxp)
close all

% plot params
si = 3; % markersize
loc = 3; % start index of orbit locations
c = linspace(1,10,maxp); % colormap

tikind = 1:10:length(c);
tik = c(tikind);
maxp_vec = 1:maxp;
tiklb = maxp_vec(tikind);

% plot labels
figure
hold on
axis([0 4 0 1])
set(gcf,'position',get(0,'screensize'))
xlabel('r')
ylabel('x')
t = ['L = ',num2str(L),', N_r = ', num2str(rlen), ', N_x_0 = ', num2str(x0len), '↩
    , p_{max} = ',num2str(maxp)];
title(t);

% fname = [pth,froot,num2str(L),'.csv'];
fid = fopen([pth,fname],'r');
tline = fgets(fid);
while ischar(tline)   % go thru all rows of M
    t1 = cellstr(tline);
    t2 = strjoin(t1);
    M = str2double(strsplit(t2,','));
    myperiod = M(2);    % period order
    w = M(1);   % r value

    if myperiod > 0
        myorbsize = ones(myperiod,1);
        rv = w*myorbsize;
        xv = M(loc:loc+myperiod-1);
```

```
        cv = c(myperiod)*myorbsize;
        scatter(rv,xv,si,cv,'filled')
    end
    tline = fgets(fid);
end
fclose(fid);

% save with meaningful title
cb = colorbar('YTick',tik,'YTickLabels',tiklb);
set(gca,'FontSize',15)
set(findall(gcf,'type','text'),'FontSize',15)
h = gcf;
pname = ['rlog_bif_L_',num2str(L),'.png'];
saveas (h, [pth,pname], 'png');
end
```

B.1.4 Lyapunov Exponent

```
%plot the lyapunov exponent of the logistic map
% [rmin, rmax] = interval over r
% maxj = number of exponents to compute
% x0 = initial condition
% q = flag for random map; if 1 then use random, else use det map

function [r,lambda] = lyapunov( rmin, rmax, maxj, x0, L, N, q )
close all
tol = 10e-6;
r = linspace(rmin+tol,rmax-tol,maxj);
r=r';
lambda = zeros(maxj,1);
n = 1000;
k = 1:N;
if q == 1 %random map
    xlen = 0;
    for j = 1:maxj
        [a,b] = myrand(L,N,r(j));
        [xv, ~, ~] = cobweb(x0, n-1, a, b, r(j), N, xlen);

        %compute xi'(x) and xi(x)
        Dxi = zeros(n,1);
        xi = zeros(n,1);
        for ic = 1:n
            xtmp = xv(ic)*ones(1,N);
            res = 2*(-a'.*2*pi.*k.*sin(2*pi*k.*xtmp) + b'.*2*pi.*k.*cos(2*pi*k.*↩
                xtmp));
            Dxi(ic) = sum(res) ;
            xi(ic) = log(r(j)) + sum(2*(a'.*cos(2*pi*k.*xtmp) - b'.*sin(2*pi*k.*↩
                xtmp)));
        end

        %compute R'(x) and R(x)
        DR = exp(xi).*Dxi;
        R = exp(xi);
```

```
        %compute f'(x)
        Df = DR.*xv.*(1-xv) + R.*(1-2*xv);
        lambda(j) = sum(log(abs(Df)))/n;
    end

else %deterministic map
    for j = 1:maxj
        x = zeros(n+1,1);
        x(1) = x0;
        for iter = 1:n-1
            x(iter + 1) = r(j) * x(iter) * (1 - x(iter));
        end
        lambda(j) = sum(log(abs(r(j) - 2*r(j)*x)))/n;
    end
end
plot(r, lambda, 'k.','MarkerSize',3)
axis([rmin rmax -1 1])
hold on
plot(r, zeros(maxj,1),'k')
set(gca,'fontsize',16)
xlabel('r','FontSize',16)
ylabel('\lambda','FontSize',16)
h = gcf;
pth = 'C:\Users\amy\Dropbox\thesis\logistic_map_code\figures\';
pname = ['rlog_lyap_L_',num2str(L),'.png'];
saveas (h, [pth,pname], 'png');
end
```

B.1.5 Period Distribution

```
% % prob of orbit

clear ; clc; close all
L = linspace(0,1,11);
L(3:end) = L(2:10);
L(1) = 0.025;
L(2) = 0.05;
L = [L(1:3) L(5) L(7) L(9) L(11)];
N = 100*ones(length(L),1);
N(1) = 400;
N(2) = 200;
maxp = 100;
tol = 10e-6;
r = linspace(.5,3.9,6);
xrng = 10;
pth = 'C:\Users\amy\Dropbox\thesis\logistic_map_code\figures\histograms\halfsig\'↩
    ;
numiters = 500;
numsims = xrng*numiters;

for i = 1:length(L)
    for j = 2:length(r)
        [data,sigma,alpha] = prob_of_orbits(L(i),N(i),r(j),xrng,numiters,maxp);
        fname = ['rlog_hist_hs_L_',num2str(L(i)),'_r_',num2str(r(j)),...
```

```
            '_s_',num2str(sigma),'_a_',num2str(alpha),'_sims_',num2str(numsims)];
        dlmwrite([pth,fname,'.csv'], data, '-append');
        plot_prob_of_orbits(data, pth, fname, L(i), r(j), maxp)
    end
end



% compute the probability of a P1, P2, etc. orbit
% L = correlation length
% N = num Fourier modes
% r = constant in [0,4]
% xrng = number of initial conditions
% maxp = max number of orbit periods

function [p,sigma,alpha] = prob_of_orbits(L, N, r, xrng, numiters, maxp)
tol = 10e-6;
xlen = maxp+5;    % number of results to print
iter = 1000;    % number of x values in the cobweb

maxsize = ceil(0.5*maxp*numiters);
% st = linspace(0+tol,1-tol,xrng);    % vector of initial conditions
st = rand(xrng,1);
u = 1;
y = zeros(maxp,2); % histogram data
p = zeros(maxp,3); %[period, avg_num, stderr]
orbits = ones(maxsize, maxp + 1)*-1;    % as many rows as periods and as many ↩
    cols as x vals

for i = 1:numiters
    disp(['iteration: ',num2str(i)])
    [a, b, sigma, alpha] = myrand(L,N,r);

    % iterate thru init conds. find periodic orbits
    for j = 1:xrng
        [xv, ~, ~] = cobweb(st(j), iter, a, b, r, N, xlen);
        for w = 1:maxp    % check period of orbit
            if abs(xv(end-w) - xv(end)) < tol
                orbits(u,1) = w;    % w is period order
                orbits(u,2) = xv(end);        % xv has orbit location

                for v = 3:w+1    % get higher order period orbit locations
                    orbits(u,v) = xv(end-v+2);
                end

                u = u + 1;    % row counter
                if u > maxsize
                    disp('u > maxsize')
                end
                break
            end
        end

    end
end

% get number of unique P1,P2... orbits for this map
```

```
u_orbits = uorbits( orbits, tol );
if isempty(u_orbits)
    p = [];
else
    num_periods = u_orbits(1:end,1); % all periods observed in col 1
    for w = 1:maxp
        y(w,1) = w;
        y(w,2) = sum(num_periods == w); % count period w orbits found
    end

    total_orbs = xrng*numiters;%sum(y(:,2))
    labels = 1:maxp;
    p(:,1) = labels';
    p(:,2) = y(:,2)/total_orbs;
    p(:,3) = sqrt((1/(total_orbs-1))*(std(p(:,2)))^2);

    % for i = 1:maxp
    %     p(i,1) = mean(y(i,2,:));
    %     p(i,2) = sum(y(i,2,:));
    % end
    % mytotal = sum(p(:,2));
    % probs = p(:,2)./mytotal;
    % mydata = [labels' probs];
end
end



% plotting orbit histogram
% data = [period, avg_num, stderr]
function plot_prob_of_orbits(data, pth, fname, L, r, maxp)
close all
bar(data(:,1),data(:,2),'g')
hold on
h = errorbar(data(:,1),data(:,2),data(:,3),'k');
set(h(1),'linestyle','none');
title(['L = ',num2str(L),' r = ',num2str(r)])
xlabel('Period')
ylabel('Average Number of Period p Orbits')
xlim([0 maxp])
set(gca,'FontSize',15)
set(findall(gcf,'type','text'),'FontSize',15)
set(gcf,'position',get(0,'screensize'))
fh = gcf;
saveas (fh, [pth,fname,'.png'], 'png');
end
```

B.2 Circle Map

B.2.1 Iterating the Circle Map

```
% Arnold circle map
```

```
% Amy Le
% Oct 20, 2014
% k (coupling strength) is simulation parameter (constant), 0<k<1
% omega (driving phase) is random param

function [xt,t,mymap] = arnold(x0, iter, a, b, k, w, N, trunc,s)
kk = 1:N;

if s ~= 1
    % smooth function
    t = x0;
    myW = zeros(1,length(t));
    for j = 1:length(t)
        xtmp = t(j)*ones(1,N);
        res = 2*(a'.*cos(2*pi*kk.*xtmp) - b'.*sin(2*pi*kk.*xtmp));
        myW(j) = exp( log(w) + sum(res) );
    end
    mymap = mod(t + myW - (k/(2*pi)) * sin(2.0 * pi * t),1);
    xt = 0;
else %s = 1
    % % bifurcation diagram calcs
    x = zeros(iter,1);
    x(1) = x0;
    for j = 1:iter-1
        xtmp = x(j)*ones(1,N);
        res = 2*(a'.*cos(2*pi*kk.*xtmp) - b'.*sin(2*pi*kk.*xtmp));
        myW = exp( log(w) + sum(res) );
        x(j+1) = x(j) + myW - (k/(2*pi)) * sin(2.0 * pi * x(j));
        x(j+1) = mod(x(j+1),1.0);
    end

    % % return the last trunc iterates only (remove transients)
    if trunc ~=0
        xt = x(end-trunc+1:end);
    else
        xt = x;
    end
    t = 0;
    mymap = 0;
end

end



% removed function call to S.m 2/19/15

function [av, bv] = myrand(L,N,alpha)

a = zeros(N,1);
b = zeros(N,1);

for j = 1:N
%     stddev = sqrt(alpha * exp(-L * j));
%     pd = makedist('Normal','mu',0,'sigma',stddev);
    Mn = sqrt(1.5*(alpha * exp(-L * j)));
    pd = makedist('Uniform','lower',-Mn,'upper',Mn);
```

```
    a(j) = random(pd);
    b(j) = random(pd);
end

av = a;
bv = b;

end
```

B.2.2 Upper and Lower Bounds on the Map

```
% bound the map at each position
% L = 0.1;   % max length
% N = 100;   % max number of modes
% w = 0.4;
% k = 1;
% xlen = 25;    % number of results to print

function envelope(k,w,alpha,N,L)

iter = 1000;    % number of x locations
kmax = 500;    % number of different realizations
st = linspace(0,1,kmax);    % vector of initial conditions
trunc = 0;

myextrema = zeros(kmax,3);
myextrema(:,1) = st';    %col1 = x position in [0,1]
myextrema(:,2) = ones(kmax,1);   %col2 = min, col3 = max
allmydata = zeros(kmax,kmax);

for kind = 1:kmax
    [a,b] = myrand(L,N,alpha);    % make a new random draw for ea realization
    [~, ~,mymap] = arnold(st, iter, a, b, k, w, N, trunc,1);
    allmydata(kind,:) = mymap;
    if kind == ceil(kmax/2)
        plot(st,mymap,'r.','Markersize',10)
        hold on
    elseif kind == ceil(kmax*(2/3))
        plot(st,mymap,'r.','Markersize',10)
    end
    clear mymap
end

myextrema(:,2) = min(allmydata)';   %black
myextrema(:,3) = max(allmydata)';   %blue
plot(myextrema(:,1),myextrema(:,2),'k.',myextrema(:,1),myextrema(:,3),'b.','↩
    markersize',10)
xlabel('x_n')
ylabel('x_{n+1}')
axis([0 1 0 1])
set(gca,'FontSize',15)
set(findall(gcf,'type','text'),'FontSize',15)
h=gcf;
```

```
name = ['envelope_unif_',num2str(kmax),'_k',num2str(k),'_L',num2str(L),'_w',↩
    num2str(w),'.png'];
path = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\';
saveas (h, [path,name], 'png');

end
```

B.2.3 Arnold Tongues Diagram

```
% tongues_driver
clear; clc; close all

tol = 10e-6;
L = [0.025 0.05 0.1 0.3 0.5 0.9];
% tongues(kmin, kmax, wmin, wmax, nstep, L, N)
tongues(0, 1.5, 0, 1, 1000, 0.025, 100)
close all
tongues(0, 1.5, 0, 1, 1000, 0.05, 100)
for i = 1:length(L)
    close all
    tongues(0, 1.5, 0, 1, 1000, L(i), 100)
end

lyap_circ_driver_k
lyap_circ_driver

% L = [0.025 0.05 0.1 0.3 0.5 0.7 0.9];
% n = 1000;
% % pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\tongues\half_alpha↩
    \';
% pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\tongues\halfa\';
% pmax = 100;
% for i = 1:length(L)
%     fname = ['tongues_u_halfa_',num2str(n),'_L_',num2str(L(i))];
%     plot_tongues(pth,fname,pmax,n)
% end
```

```
% arnold tongues picture
% Amy Le, Feb 19, 2015
% in:
% [kmin, kmax] = range over k values
% [wmin, wmax] = range over omega values
% nsteps = create an nstep x nstep grid for display

function tongues(kmin, kmax, wmin, wmax, nstep, L, N)

tol = 0.0001;
pmax = 100;   % max period
x0 = zeros(nstep,nstep);    % x has nstep init conditions, all same
period = pmax - zeros(nstep,nstep);   % all periods have order 100
omegavect = linspace(wmin, wmax, nstep);
```

```
kvect = linspace(kmin,kmax,nstep);

% random draws
alpha = .5*(10e-5);
if N < (10/L)
    N = ceil(10/L);
end
[a, b] = myrand(L,N,alpha);
iter = 1200;
trunc = 150;   % will make arnold return a vec 150 units long

% iterate the map, remove transients
% go thru each ic in x0 and each k,w value
% arnold returns a vector xt of final postions
% look for period order in xt
for i = 1:nstep
    disp(['tongue iter: ',num2str(i)])
    for j = 1:nstep
        [xv,~,~] = arnold(x0(i,j), iter, a, b, kvect(i), omegavect(j), N, trunc↩
            ,1);
        if isnan(xv(end)) == 0
            for w = 1:pmax    % check period of orbit
                if abs(xv(end-w) - xv(end)) < tol
                    period(i,j) = w;    % w is period order; default period is ↩
                        pmax
                    break
                end
            end
        end

    end
end


% set up figure
set(gcf,'position',get(0,'screensize'))
period = flipud(period);   %reflect image to get proper orientation
image(period)   ;

% custom colormap
c1 = colorcube(pmax);
colormap([c1(1:ceil(pmax/2),:); jet(ceil(pmax/2)-1) ;0*white(1)]);
cb = colorbar;

% axis labels
xlabel('\omega')
set(gca,'XTickMode', 'Manual');
oind = 0:ceil(nstep/10):nstep;
oind = oind(2:end);
strom = strsplit(sprintf('%1.1f ',omegavect(oind)));
set(gca,'XTickLabel',strom)

set(gca,'YLim', [0 nstep])
ylabel('k')
set(gca,'YTickMode', 'Manual');
kind = 0:ceil(nstep/10):nstep;
kind = kind(2:end);
strk = strsplit(sprintf('%1.1f ',fliplr(kvect(kind))));
```

```
set(gca,'YTickLabel',strk)
set(gca,'FontSize',15)
set(findall(gcf,'type','text'),'FontSize',15)

% save a data and png file
h = gcf;
name = ['tongues_u_halfa_',num2str(nstep),'_L_',num2str(L),'.png'];
% 'C:\Users\swamy\Documents\amy\thesis\circle_map_code\circle_map_code\figures\↩
    normal_tongues\';
pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\tongues\halfa\';
dlmwrite([pth,['tongues_u_halfa_',num2str(nstep),'_L_',num2str(L),'.csv']],period↩
    )
saveas (h, [pth,name], 'png');
end
```

B.2.4 Lyapunov Exponent

```
% plot the lyapunov exponent of the circle map
% input
% [wmin, wmax] = min and max values of omega, e.g. [0,1]
% maxj = number of lyapunov exponents to find
% x0 = initial condition for the map
% k = parameter, chaos is observed for k>1
% q = random map flag; if q == 1 then use random map, else, use det map
% L = correlation length
% N = number of fourier modes

% output
% plot of the lyapunov exponent vs omega
function lambda = lyapunov_circ( wmin, wmax, maxj, x0, k, L, N, alpha, q )
close all
kk = 1:N;
om = linspace(wmin,wmax,maxj);
lambda = zeros(maxj,1);
n = 1000;
if q == 1 %random map
    xlen = 0;
    for j = 1:maxj
        [a,b] = myrand(L,N,alpha);
        [xv, ~, ~] = arnold(x0, n, a, b, k, om(j), N, xlen, q);

        %compute xi'(x) and xi(x)
        Dxi = zeros(n,1);
        xi = zeros(n,1);
        for ic = 1:n
            xtmp = xv(ic)*ones(1,N);
            res = 2*(-a'.*2*pi.*kk.*sin(2*pi*kk.*xtmp) + b'.*2*pi.*kk.*cos(2*pi*k↩
                .*xtmp));
            Dxi(ic) = sum(res) ;
            xi(ic) = log(om(j)) + sum(2*(a'.*cos(2*pi*kk.*xtmp) - b'.*sin(2*pi*kk↩
                .*xtmp)));
        end

        %compute Omega'(x)
```

```
        DW = exp(xi).*Dxi;

        %compute f'(x)
%         Df = DR.*xv.*(1-xv) + R.*(1-2*xv);
        Df = 1 - k * cos(2.0 * pi * xv) + DW;
        lambda(j) = sum(log(abs(Df)))/n;
    end
    pname = ['rcirc_u_halfa_lyap_',num2str(maxj),'_L_',num2str(L),'_k_',num2str(k↩
        ),'_w.png'];
else
    for j = 1:maxj
        x = zeros(n,1);
        x(1) = x0;

        for iter = 1:n-1
            x(iter + 1) = x(iter) + om(j) - (k/(2*pi)) * sin(2.0 * pi * x(iter));
        end
        lambda(j) = sum(log( 1 - k * cos(2.0 * pi * x)))/n;
    end
    pname = ['detcirc_lyap_',num2str(maxj),'_k_',num2str(k),'_w.png'];
end
plot(om, lambda, 'k.','MarkerSize',3)
axis([wmin wmax -1 1])
hold on
plot(om, zeros(maxj,1),'k')
set(gca,'fontsize',16)
xlabel('\omega','FontSize',16)
ylabel('\lambda','FontSize',16)
title(['k = ',num2str(k)],'FontSize',16)
h = gcf;
pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\lyapunov\uniform\halfa↩
    \';
% pth = 'C:\Users\swamy\Documents\amy\thesis\circle_map_code\circle_map_code\↩
    figures\lyapunov\';
saveas (h, [pth,pname], 'png');
end



% plot the lyapunov exponent of the circle map
% input
% [wmin, wmax] = min and max values of omega, e.g. [0,1]
% maxj = number of lyapunov exponents to find
% x0 = initial condition for the map
% k = parameter, chaos is observed for k>1
% q = random map flag; if q == 1 then use random map, else, use det map
% L = correlation length
% N = number of fourier modes

% output
% plot of the lyapunov exponent vs omega
function lambda = lyapunov_circ_k( kmin, kmax, maxj, x0, w, L, N, alpha, q )
close all
kk=1:N;
k = linspace(kmin,kmax,maxj);
lambda = zeros(maxj,1);
n = 1000;
```

```
if q == 1 %random map
    xlen = 0;
    for j = 1:maxj
        [a,b] = myrand(L,N,alpha);
%         arnold(x0, iter, a, b, k, w, N, trunc,s)
        [xv, ~, ~] = arnold(x0, n, a, b, k(j), w, N, xlen, q);

        %compute xi'(x) and xi(x)
        Dxi = zeros(n,1);
        xi = zeros(n,1);
        for ic = 1:n
            xtmp = xv(ic)*ones(1,N);
            res = 2*(-a'.*2*pi.*kk.*sin(2*pi*kk.*xtmp) + b'.*2*pi.*kk.*cos(2*pi*↩
                kk.*xtmp));
            Dxi(ic) = sum(res) ;
            xi(ic) = log(k(j)) + sum(2*(a'.*cos(2*pi*kk.*xtmp) - b'.*sin(2*pi*kk↩
                .*xtmp)));
        end

        %compute Omega'(x)
        DW = exp(xi).*Dxi;

        %compute f'(x)
%         Df = DR.*xv.*(1-xv) + R.*(1-2*xv);
        Df = 1 - k(j) * cos(2.0 * pi * xv) + DW;
        lambda(j) = sum(log(abs(Df)))/n;
    end
    pname = ['rcirc_u_halfa_lyap_',num2str(maxj),'_L_',num2str(L),'_w_',num2str(w↩
        ),'_k.png'];
else
    for j = 1:maxj
        x = zeros(n,1);
        x(1) = x0;

        for iter = 1:n-1
            x(iter + 1) = x(iter) + w - (k(j)/(2*pi)) * sin(2.0 * pi * x(iter));
        end
        lambda(j) = sum(log( 1 - k(j) * cos(2.0 * pi * x)))/n;
    end
    pname = ['detcirc_u_lyap_',num2str(maxj),'_w_',num2str(w),'_k.png'];
end
plot(k, lambda, 'k.','MarkerSize',3)
axis([kmin kmax -1 1])
hold on
plot(k, zeros(maxj,1),'k')
set(gca,'fontsize',16)
xlabel('k','FontSize',16)
ylabel('\lambda','FontSize',16)
title(['\omega = ',num2str(w)],'FontSize',16)
h = gcf;
pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\lyapunov\uniform\halfa↩
    \';
% pth = 'C:\Users\swamy\Documents\amy\thesis\circle_map_code\circle_map_code\↩
    figures\lyapunov\';
saveas (h, [pth,pname], 'png');
end
```

B.2.5 Devil's Staircase and Kernel Density Estimation

```
% ------------------------------------------------------------------
% generate a bunch of staircases
% L = correlation length
% k = set the constant coupling strength parameter
% n = number of om to test in [0,1]

clear; clc; close all;

L = [0.05 0.1 0.5 1 1.5];
% L=[0.05 1.5];
k = [1.5 0.7 1];
n = 10000;
for i = 1:length(L)
    for j = 1:length(k)
        staircase_driver(L(i),k(j),n)
    end
end


% ------------------------------------------------------------
% histogram of rotation number over 1000 realizations for fixed om

% k = 1;
% L = 0.1;
% n = 1000;     % number of realizations
% x0 = 0.1;    % initial condition
% N = ceil(10/L);    % number of Fourier modes
% iter = 1000;    % number of times the cobweb iterates
% alpha = 10e-5;    % param for random parameters (myrand)
% om = 0.225;
% rho = zeros(n,1);
%
% for i = 1:n
%     [a,b] = myrand(L,N,alpha);
%     rho(i) = staircase(N, om, k, iter, x0, a, b);
% end
%
% s = num2str(n);
% t = strcat(s, ' Simulations');
% figure
% h = gcf;
% % hist(rho)
% [f,xi] = ksdensity(rho);
% plot(xi,f)
% ylim([0 n])
% tmp = xlim;
% text(tmp(1)+0.01,n*0.75,['\fontsize{10}L=',num2str(L),', \fontsize{10}k=',↩
    num2str(k),', \fontsize{10}\omega=',num2str(om)])
% xlabel('\rho')
% ylabel('frequency')
% title(['Kernel Density Estimator of Rotation Numbers;',t])
% name = ['kde_rho_k',num2str(k),'_L',num2str(L),'_om',num2str(om),'.png'];
% path = 'C:\Users\amy\Dropbox\thesis\circle map code\figures\';
```

```
% saveas (h, [path,name], 'png');
%



% devil's staircase
% plot the roation number rho as a function of omega
% L = correlation length
% k = set the constant coupling strength parameter
% n = number of om to test in [0,1]

function staircase_driver(L,k,n)
close all;
x0 = 0.1;    % initial condition

N = ceil(10/L);    % number of Fourier modes
if N < 100
    N = 100;
end
iter = 1000;    % number of times the cobweb iterates
alpha = 10e-5;    % param for random parameters (myrand)
om = linspace(0,1,n);

[a,b] = myrand(L,N,alpha);
rho = zeros(length(om),1);
for i = 1:length(om)
    rho(i) = staircase(N, om(i), k, iter, x0, a, b);
    %         dlmwrite(fname, res, '-append');
end
figure
plot(om,rho,'k.','markersize',3)
axis([min(om) max(om) -.10  1.1])

set(gca,'fontsize',16)
xlabel('\omega','FontSize',16)
ylabel('\rho','FontSize',16)
title(['k = ',num2str(k)],'FontSize',16)

% text(.1,.9,['\fontsize{10}L=',num2str(L)])

h = gcf;
t = strcat('rcirc_u_devil_k',num2str(k));
p = strcat('_L',num2str(L));
name = strcat([t,p],'.png');
% path = 'C:\Users\swamy\Documents\amy\thesis\circle map code\figures\';
% path = 'C:\Users\amy\Dropbox\thesis\presentation\images\';
path = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\devil\uniform\';
saveas (h, [path,name], 'png');
set(gca,'FontSize',15)
end



% get location of orbits for bifurcation diagram
% rho = (final position - start position after removing transients) /
% numSteps
```

```
function rho = staircase(N, w, k, iter, x0, a, b)
x = x0;
kk = 1:N;
for i = 1:floor((iter-1)/2) % transients
    xtmp = x*ones(1,N);
    res = 2*(a'.*cos(2*pi*kk.*xtmp) - b'.*sin(2*pi*kk.*xtmp));
    myW = exp( log(w) + sum(res) );
    x = x + myW - (k/(2*pi)) * sin(2.0 * pi * x);
end
xin = x;
for i = 1:floor((iter-1)/2) % output
    xtmp = x*ones(1,N);
    res = 2*(a'.*cos(2*pi*kk.*xtmp) - b'.*sin(2*pi*kk.*xtmp));
    myW = exp( log(w) + sum(res) );
    x = x + myW - (k/(2*pi)) * sin(2.0 * pi * x);
    rho = (x-xin)/i;    %rho changes each time until end of loop
end
end
```

B.2.6 Period Distribution

```
% % % prob of orbit

clear; clc; close all
pth = 'C:\Users\amy\Dropbox\thesis\logistic_map_code\figures\histograms\maxsig\';
fname = 'rlog_hist_L_0.1_r_3.3_s_0.017565_a_1.5414e-05_sims_5000';
% pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\histogram\↩
    normal_maxa\';
% fname = 'rcirc_hist_n_L_01_w_06_k_1_sims_5000';
data = csvread([pth,fname,'.csv']);
plot_prob_of_orbits(data, pth, fname, 0.1, 0.6, 1, 100)
% L = linspace(0,1,11);
% L(3:end) = L(2:10);
% L(1) = 0.025;
% L(2) = 0.05;
% L = [L(1:3) L(5) L(7) L(11)];
% N = 100*ones(length(L),1);
% N(1) = 400;
% N(2) = 200;
% maxp = 100;
% tol = 10e-6;
% alpha = (10e-5);
%
% w = [.4,.6,.9];
% xrng = 10;
% pth = 'C:\Users\amy\Dropbox\thesis\circle_map_code\figures\histogram\↩
    normal_maxa\';
% numiters = 500;
% numsims = xrng*numiters;
% k = [1,1.5];
% for kk = 1:length(k)
%     disp(['k: ',num2str(k(kk))])
%     for i = 1:length(L)
```

```
%         disp(['L: ',num2str(L(i))])
%         for j = 2:length(w)
%             disp(['w: ',num2str(w(j))])
%             data = prob_of_orbits(L(i),N(i),alpha,w(j),k(kk),xrng,numiters,maxp↩
    );
%             fname = ['rcirc_hist_n_L_',num2str(L(i)),'_w_',num2str(w(j)),...
%                 '_k_',num2str(k(kk)),'_sims_',num2str(numsims)];
%             dlmwrite([pth,fname,'.csv'], data, '-append');
%             plot_prob_of_orbits(data, pth, fname, L(i), w(j),k(kk), maxp)
%         end
%     end
% end



% compute the probability of a P1, P2, etc. orbit
% L = correlation length
% N = num Fourier modes
% r = constant in [0,4]
% xrng = number of initial conditions
% maxp = max number of orbit periods

function p = prob_of_orbits(L, N, alpha, omega, k, xrng, numiters, maxp)
tol = 10e-6;
xlen = maxp+5;    % number of results to print
iter = 1000;    % number of x values in the cobweb

maxsize = ceil(0.5*maxp*numiters);
% st = linspace(0+tol,1-tol,xrng);    % vector of initial conditions
st = rand(xrng,1);
u = 1;
y = zeros(maxp,2); % histogram data
p = zeros(maxp,3); %[period, avg_num, stderr]
orbits = ones(maxsize, maxp + 1)*-1;    % as many rows as periods and as many ↩
    cols as x vals

for i = 1:numiters
    [a, b] = myrand(L,N,alpha);

    % iterate thru init conds. find periodic orbits
    for j = 1:xrng
        [xv, ~, ~] = arnold(st(j), iter, a, b, k, omega, N, xlen,1);

        for w = 1:maxp    % check period of orbit
            if abs(xv(end-w) - xv(end)) < tol
                orbits(u,1) = w;    % w is period order
                orbits(u,2) = xv(end);        % xv has orbit location

                for v = 3:w+1    % get higher order period orbit locations
                    orbits(u,v) = xv(end-v+2);
                end

                u = u + 1;    % row counter
                if u > maxsize
                    disp('u > maxsize')
                end
                break
```

```
            end
        end

    end
end

% get number of unique P1,P2... orbits for this map
u_orbits = uorbits( orbits, tol );
if isempty(u_orbits)
    p = [];
else
    num_periods = u_orbits(1:end,1); % all periods observed in col 1
    for w = 1:maxp
        y(w,1) = w;
        y(w,2) = sum(num_periods == w); % count period w orbits found
    end

    total_orbs = xrng*numiters;%sum(y(:,2))
    labels = 1:maxp;
    p(:,1) = labels';
    p(:,2) = y(:,2)/total_orbs;
    p(:,3) = sqrt((1/(total_orbs-1))*(std(p(:,2)))^2);
end
end




% plotting orbit histogram
% data = [period, avg_num, stderr]
function plot_prob_of_orbits(data, pth, fname, L, w, k, maxp)
close all

% bar(data(:,1),data(:,2),'g')
% hold on
% h = errorbar(data(:,1),data(:,2),data(:,3),'k');
% set(h(1),'linestyle','none');
% set(gca,'YScale','log')
% set(gca,'YDir','reverse')
semilogy(data(:,1),data(:,2),'k.','MarkerSize',12)
fname = [fname,'semilogy'];

title(['L = ',num2str(L),', r = 3.3'])%\omega = ',num2str(w), ', k = ',num2str(k)↩
    ])
xlabel('Period')
ylabel('Average Fraction of Period p Orbits')
xlim([0 maxp])

set(gca,'FontSize',15)
set(findall(gcf,'type','text'),'FontSize',15)
% set(gcf,'position',get(0,'screensize'))
% saveas (gcf, [pth,fname,'.png'], 'png');
saveas (gcf, ['C:\Users\amy\Dropbox\thesis\writeup\figs\',fname,'.png'], 'png');
end
```